AD-A265 116

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DTIC
ELECTE
MAY 2 8 1993
S C D

THESIS

AN INVESTIGATION USING EMPIRICAL ORTHOGONAL FUNCTIONS AND OBJECTIVE ANALYSIS TO ANALYSE THE VERTICAL TEMPERATURE STRUCTURE OF A GULF STREAM MEANDER

by

Martin J. Sauze
March 1993

Thesis Advisor: Everett F. Carter

Approved for public release; distribution is unlimited

93-12028

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 55	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS Program Element No / Project No. / Task No. / Work Unit Accession Number

11. TITLE (Include Security Classification)
AN INVESTIGATION USING EMPERICAL ORTHOGONAL FUNCTIONS AND OBJECTIVE ANALYSIS TO ANAYSE THE VERTICAL TEMPERATURE STRUCTURE OF A GULF STREAM MEANDER.

12. PERSONAL AUTHOR(S) Martin J. Sauze

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Master's Thesis	From To	March 1993	155

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)
Oceanography, Empirical orthogonal functions, Objective analysis, Reconstruction of synthetic vertical temperature profiles, error analysis of reconstruction, Gulf stream meander, Number of XBT's needed to model meander.

19. ABSTRACT (continue on reverse if necessary and identify by block number)

Expendable bathymetric temperature (XBT) data taken from an anticyclonic meander crest within the Gulf Stream (Hummon 1991) is analysed by looking at the empirical vertical structure. The ensemble averaged data is formed into a projection matrix that compares the value of the temperature at one depth with the temperature at a second depth. The data was smoothed with the correlation analysis being performed at ten metre intervals from five metres to a depth of 800 metres. The first four empirical orthogonal functions (EOF's) of the projection matrix were computed and the modal amplitudes for each XBT determined. Using objective analysis the modal amplitudes were interpolated onto a specified grid. Synthetic XBT's were then reconstructed at the grid positions using the interpolated grid modal amplitude values. A measure of the error at each grid point was determined. The objective analysis was subsequently repeated using successively fewer XBT's from the data set, until the resulting error in the interpolated XBT's at the grid points became unacceptable.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Everett F. Carter	22b. TELEPHONE (Include Area code) (408) 656-3318 22c. OFFICE SYMBOL OC/Cr

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

An Investigation using Empirical Orthogonal Functions and Objective Analysis to analyse the vertical temperature structure of a Gulf Stream meander.

by

Martin J. Sauze
Lieutenant Commander, Royal Navy
B.Ed., University of Nottingham England 1977

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN PHYSICAL OCEANOGRAPHY

from the

NAVAL POSTGRADUATE SCHOOL
March 1993

Author: M.J. Sauze
Lt Cdr Martin J. Sauze

Approved by: Everett Carter
Assistant Professor Everett F Carter, Thesis Advisor

James A Cummings
Doctor James A. Cummings, Second Reader

Curtis A Collins
Professor Curtis A. Collins, Chairman
Department of Oceanography

ABSTRACT

Expendable bathymetric temperature (XBT) data taken from an anticyclonic meander crest within the Gulf Stream (Hummon et al 1991) is analysed by looking at the empirical vertical structure. The ensemble averaged data is formed into a projection matrix that compares the value of the temperature at one depth with the temperature at a second depth. The data is smoothed with the correlation analysis being performed at 10 metre intervals from 5 metres to a depth of 800 metres. The first four, or principle, EOFs of the projection matrix are computed and the modal amplitudes for each XBT determined. Using objective analysis the modal amplitudes are interpolated onto a specified grid. Synthetic XBTs are then reconstructed at the grid positions using the interpolated grid modal amplitude values. A measure of the error variance at each grid point is determined. The objective analysis is repeated using successively fewer XBTs from the data set, until the resulting error in the interpolated XBTs at the grid points become unacceptable.

DTIC [illegible] ECTED 5

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution /

Availability Codes

Dist | Avail and / or Special
A-1

TABLE OF CONTENTS

I INTRODUCTION

From a military standpoint, to carry out a successful range prediction against a surface or subsurface unit, for either passive or active SONAR, it is necessary to have access to the most recent vertical temperature profile that is available for the area of interest. If several XBTs are taken at different positions and at different times within a region, what is the optimal vertical profile at some arbitrary point of significance within the region based upon this collected data? Or, perhaps if multiple units are on task, each taking their own XBTs, what is the optimal interpolation of the water condition at some point between the units? The development of range dependent prediction models makes a knowledge of the water conditions between a unit and its target even more crucial. Thus the ability to be able to empirically assess the vertical water conditions at any point within a target region, and to obtain valid and useful information, is of considerable importance.

From a purely scientific basis, it would be of benefit to know the approximate number of vertical profiles that need to be obtained before a comprehensive analysis of a given area could be achieved. Similarly, some measure of the optimal spacing between XBTs would be of value for planning and the economic use of valuable assets and time.

Recent developments in satellite technology now allow the determination of the subsurface vertical structure by measuring the dynamic height of the ocean using altimetry (Carnes et al 1990). But, how many readings need to be taken for a given region of the ocean before a realistic interpretation can be constructed? Additionally, if gaps exist within the data collection, how much error will exist in interpolating data void areas? If sufficient remote readings could randomly be taken in and around a given feature, how many readings would be required before the feature's vertical temperature structure can be adequately reproduced?

The ultimate goal of this study is to find out how few XBTs are required before an adequate vertical temperature profile can be compiled within a Gulf Stream meander.

The feature analyzed in this study is the warm side of a Gulf Stream meander that was identified and rigorously sampled during 1988 (Hummon et al 91). It is anticipated that a study of this type conducted in this particular area will be of general use, and give an indication of the number of XBTs that need to be deployed before an adequate interpolation can be made as to the underlying water structure.

Carter and Robinson (1987) considered empirically the effects of reducing the size of an original data set upon the value of the contour maps that were produced. They considered the depth of the 15 degree Celsius isotherm. The data were taken during the POLYMODE experiment and consisted

of 443 XBTs. The depth of the 15 degree isotherm was extracted from each of these XBTs and optimally interpolated onto a regular grid. The results of the interpolation are shown at the top left of Figure 1 with the associated amount of error, for any location, depicted in the top right of Figure 1. The data set was then halved and the analysis repeated (the middle two pictures), with the effect that the new results showed very little change in the error field. However, as the last two diagrams show, by the time only a quarter of the data is included the error in the analysis field has grown considerably, to the point where the analysis is unacceptable for practical purposes. The study indicates that in order to survey the given area of the ocean it would have been sufficient to have launched half of the XBTs that were actually launched without any serious decrease in the quality of the 15 degree isotherm map that was produced. Additionally, the interpolation procedure they employed gives an explicit statement of the amount of error involved in the reconstruction at any location within the region thereby, giving an unequivocal statement about its usefulness, or otherwise, to a future user.

The object of this study is to consider the reduction of data problem in an objective analysis procedure more rigorously. By reducing a data set repeatedly by one observation until the resulting error in the reconstructed vertical temperature profile becomes unacceptable, the minimum

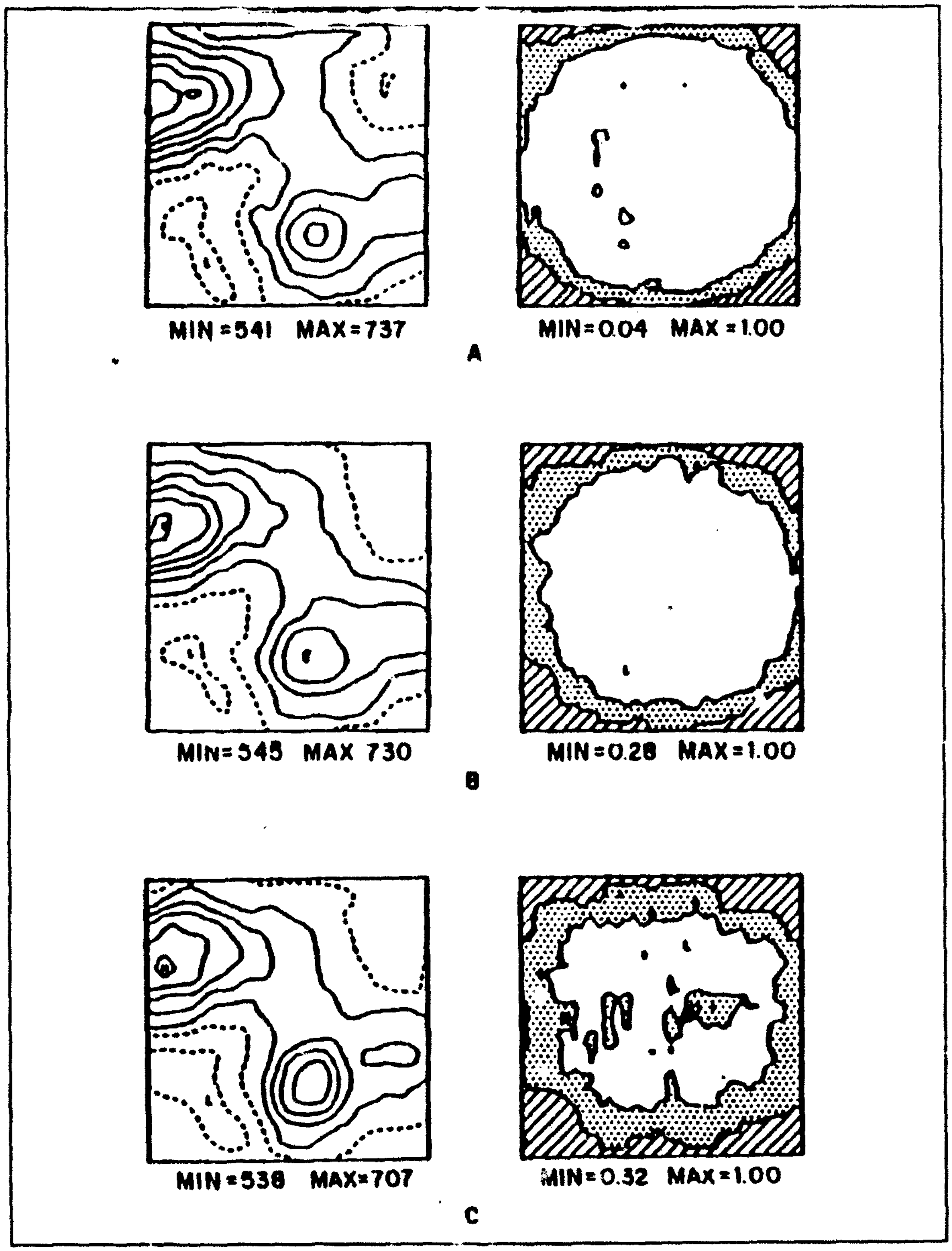

Figure 1 The result of an objective analysis of the depth of the 15 C isotherm using different amounts of data, for a six degree square centred at 70 W 29 N. 25 m contour interval for analysis, 0.25 contour interval for error. A B C represent 443 222 and 111 observations respectively (after Carter & Robinson 1987).

number of XBTs required in the analysis can be determined. The area of ocean under consideration in this study is very difficult to interpolate adequately without a large amount of data because of the high degree of variability that exists within short spatial and temporal ranges.

Rather than just looking at one parameter, such as sea surface temperature or the depth of the 15 degree isotherm, this study will seek to reproduce the vertical temperature structure at a given position within the region from the surface to a depth of 800 metres. The analysis will also give an account of the average error involved in creating a synthetic XBT profile.

There are two theoretical strands that are considered; (1) the theory of objective analysis and (2) the theory of Empirical Orthogonal Functions.

The first, objective analysis, describes a method to take a finite number of data points, at irregular spatial or temporal intervals over an area of the ocean's surface, and interpolate the data in such a manner that an optimal estimate of a scalar value can be obtained for any given location within the region.

The amount of data to be interpolated per grid point is further reduced by exploiting the properties of Empirical Orthogonal Functions. The use of EOFs allows a given XBT to be broken down into modes that are constant for the whole data set, and into corresponding modal amplitudes that are unique

uto each particular XBT. Then, for each XBT, the sum of the products of the modes and the corresponding modal amplitudes give a complete representation of the XBT in question. However, the first few EOFs often explain the majority of the structure of the complete XBT. Thus it is possible to approximately reconstruct each XBT with a reduction in the data. If, for instance, only the first 4 modes are considered, then each XBT is represented by just 4 unique numbers, the modal amplitudes.

Having determined the unique modal amplitudes for each XBT, objective analysis is used to optimally interpolate the four principle sets of numbers onto a regular grid. Multiplying each in turn by its corresponding mode, results in a synthetic XBT being reconstructed at each of the grid points.

Having synthetically produced XBTs at each grid point, an objective error analysis is used to estimate the total error variance of each of the synthetic XBTs. Thereafter, using a random generator, successive XBTs are removed from the original data set. The objective analysis and reconstruction are repeated until the error variance in the synthetic XBTs become unacceptable.

II THEORY

A. INTRODUCTION

This chapter is divided into 4 main sections. The first section outlines the development and theory of EOFs and gives an account of the use of EOFs in oceanography. Similarly, section two covers the background of objective analysis and is followed by a development of the theory. The third section explains how the error analysis of the modal amplitudes is used to account for the error in the reconstructed synthetic XBT. The final section outlines how all the strands can be brought together.

B. DEVELOPMENT AND THEORY OF EMPIRICAL ORTHOGONAL FUNCTIONS

1. Development of Empirical Orthogonal Functions

The theory of Principle Component Analysis was first proposed by Pearson (1901) and developed into a comprehensive theory by Hotelling (1933). Hotelling's work led Kelly (1935) to advance a model suitable for modern computer usage. The theory was first put into practice by Wrigley and Nechus (1955) in the field of psychology.

Lorenz (1956) outlined the theoretical basis for the use of Principle Component analysis in meteorology,

demonstrating its use as an aid to efficient weather prediction and coining the phrase Empirical Orthogonal Functions (EOFs) which has become the accepted norm within the geophysical sciences. The value of EOFs as a tool in geophysical research is reflected in the variety of uses to which they have been put. For instance, in meteorological research, which requires working with large data sets, the use of EOFs have been used to reduce the volume of data that need to be interpolated or stored.

Stidd (1966) used EOFs to study climatological rain fall patterns within the State of Nevada. By interpolating EOF analysis between climate stations he was able to successfully reconstruct the climate record of a station that had been removed from the initial analysis. This is similar to the current study in that the temperature data at each XBT site, like the rain fall data, is represented by modal amplitudes, and the data must be interpolated to additional locations using objective analysis.

EOFs have been featured highly in climatological studies causing Mitchell (1966) to comment that EOFs may be of significant use as climatological indicators. This view is strengthened by the work of researchers like Kutzbach (1967), who used EOF analysis successfully to combine climatological records of temperature, precipitation and surface pressure over the United States; and Kidson (1974), who used EOFs to

produce climatological indicators for both hemispheres and the tropics.

Paegle and Haslam (1982) used EOFs in the prediction of the 500 and 850 mb pressure heights over a 24 hour period. Wallace and Dickinson (1972) showed how EOFs may be applied to time series analysis, reducing the data processing required and increasing the efficiency for spectral modelling of the atmosphere.

In oceanography the technique is finding an increasingly wide variety of uses. For instance Kundu (1975) used EOFs in a time series analysis of velocity fields along the Oregon coast. Carnes et al (1990) have shown that EOFs can be used in conjunction with satellite derived ocean dynamic heights to obtain a measure of the ocean's subsurface vertical temperature structure. Oceanographic models at Fleet Numerical Oceanographic Center (FNOC), such as the Optimal Thermal Interpolation System (OTIS), employ EOFs to effectively represent ocean thermal climotologies (Tunnicliffe and Cummings 1991). Similarly, the Navy/NOAA Oceanographic Data Distribution System (NODDS) includes the use of EOF techniques to compress large volumes of data, enabling distant users either ashore or at sea to receive by telephone link sophisticated real time ocean and meteorological information using a desk top PC.

2. Theory of Empirical Orthogonal Functions's

The above examples show the versatility and value of EOFs as an effective tool within the fields of meteorology and oceanography. Set out below is a development of the basic theory. The approach outlined considers the work of Lorenz (1956), who first described the use of EOFs in geophysical research, Harman (1976), who formally derives the general theory, and Dunteman (1989), whose clarity and examples gave considerable insight into the technique.

The object of Principle Component Analysis is to take a large body of data and empirically reduce it. The model assumes a linear set of numbers such that a linear combination of these components leads to a complete representation of the original data set.

Mathematically the method assumes that,

$$P_i = q_1 y_1 + q_2 y_2 + \ldots\ldots\ldots q_j y_j$$

equation 1

where (j = 1,2,......n) and each of the observed variable P_i is described linearly in terms of n orthogonal components $y_1, y_2, \ldots y_n$. The power of this approach being that only a few of the components need to be retained in order to retain the majority of the total variance.

The coefficients q_j, are referred to as the "loadings, "scores" or "weightings" and in geophysics as "modal amplitudes". Each modal amplitude is multiplied by its corresponding principle component, with the sum being equal to the value of the original variable. The problem is to find suitable values of q and y to be able to represent the variable p_i in question. This is efficiently achieved by expressing equation 1 in matrix form as,

$$\boldsymbol{P}=\boldsymbol{QY}$$

equation 2

where $\boldsymbol{P}$ is an m by n matrix of scalar variables whose columns represent the vector P_i, $\boldsymbol{Q}$ is an m by n matrix of modal amplitudes and $\boldsymbol{Y}$ represents n column vectors each with m rows.

Consider the situation where m elements $(P_i i=1...m)$ have been measured at n different locations. For this study, 80 isotherm depth measurements (m) made at each of 156 locations (n).

Let

$$\boldsymbol{A}=\boldsymbol{p}^{*\prime}\boldsymbol{p}^{*}$$

equation 3

where

$$P^{*}=P_{i}-\overline{P}$$

equation 4

the difference of a value P_i from the mean value $\overline{P}$. $\boldsymbol{P}^{*\prime}$ is the transpose of $\boldsymbol{P}^{*}$ and $\boldsymbol{A}$ represents the covariance matrix formed by the dot product of $\boldsymbol{P}^{*\prime}$ and $\boldsymbol{P}^{*}$. The covariance matrix $\boldsymbol{A}$ is normalized to form the correlation matrix $\boldsymbol{A}$,

$$\boldsymbol{A}=\langle \boldsymbol{P}^{*\prime}\rangle\langle \boldsymbol{P}^{*}\rangle$$

equation 5

and the symbols <> denote an ensemble averaging of the variance from each data point. The matrix $\boldsymbol{A}$ is also known as the projection matrix.

From the theory of matrix algebra (Harman 1976) a general matrix $\boldsymbol{G}$ can be expressed in terms of its eigenvectors $\boldsymbol{e}$ and eigenvalues $\boldsymbol{\lambda}$ such that,

$$\boldsymbol{G}e=\boldsymbol{\lambda} e$$

equation 6

where $\boldsymbol{Ge}$ is regarded as a transformation of $\boldsymbol{e}$ with λ as the constant of proportionality. Each root λ_i has a non zero solution e_i and the m roots $\lambda_1, \lambda_2, \ldots\ldots\ldots \lambda_m$ lead to n values $e_1, e_2 \ldots\ldots e_n$ such that equation 6 may be written as,

$$\boldsymbol{G}(e_1, e_2 \ldots . e_n) = \lambda_1 e_1, \lambda_2 e_2 \ldots . \lambda_n e_n$$

equation 7

or in matrix form

$$\boldsymbol{GE} = \boldsymbol{E}\Lambda$$

equation 8

where $\Lambda = diag(\lambda_1, \lambda_2 \ldots . \lambda_n)$.

Inserting matrix $\boldsymbol{A}$ from equation 5 into the general equation 8 gives,

$$\boldsymbol{AY} = \lambda \boldsymbol{Y}$$

equation 9

The vectors y_i are linearly independent such that the determinants are non zero and $\boldsymbol{Y}$ has a unique inverse $\boldsymbol{Y}^{-1}$,

$$\boldsymbol{Y}^{-1}\boldsymbol{AY} = \Lambda.$$

equation 10

Since $\boldsymbol{A}$ is a correlation matrix and is symmetric

$$\boldsymbol{A}=\boldsymbol{A}'$$

equation 11

$(\boldsymbol{Y}^{-1})'$ are characteristic values with $\boldsymbol{Y}$ being orthogonal with the property that,

$$\boldsymbol{Y}\boldsymbol{Y}'=\boldsymbol{I}$$

equation 12

the identity matrix or,

$$\boldsymbol{Y}^{-1}=\boldsymbol{Y}'$$

equation 13

giving that equation 14 can be written as,

$$\boldsymbol{Y}'\boldsymbol{A}\boldsymbol{Y}=\Lambda$$

equation 14

Equation 14 states that the symmetric matrix $\boldsymbol{A}$ may be diagonalized by means of the orthogonal transformations $\boldsymbol{Y}$ and that the elements of Λ and $\boldsymbol{Y}$ are real with $\boldsymbol{Y}$ being made up of n characteristic linearly independent equations.

From equation 14 diagonally decomposing $\boldsymbol{A}$ gives values for matrix $\boldsymbol{Y}$, and Λ. Now knowing the values of $\boldsymbol{P}$ and $\boldsymbol{Y}$, $\boldsymbol{Q}$, the matrix of coefficients or modal amplitudes, can be determined from equation 2. Having obtained values for $\boldsymbol{Q}$ and $\boldsymbol{Y}$ equation 2 states that the value of $\boldsymbol{P}$ can be exactly determined and that it equals the matrix product of $\boldsymbol{Q}$ and $\boldsymbol{Y}$.

From equations 2,13,and 14 it follows (Paegle and Haslam 1982) that the total variance is given by the sum of the eigenvalues,

$$\sum_i \overline{p^{*2}} = \sum_i \lambda$$

equation 15

and that each eigenvalue λ_i gives the contribution of each eigenvector $\boldsymbol{Y}_i$ to the total variance of $\boldsymbol{P}$.

When the eigenvalues are arranged in descending order the variance represented by each mode or eigenvector decreases dramatically as the number of the eigenvalues are increased. A realistic estimate of the original data $\hat{\boldsymbol{P}}$ can thus be achieved by using only the first few modes $\hat{\boldsymbol{Y}}$ and the corresponding modal amplitudes $\hat{\boldsymbol{Q}}$.

$$\hat{P}=\hat{Q}\hat{Y}'.$$

equation 16

The use of a limited number of modes reduces the quantity of data that has to be stored and processed. In addition the variability in the higher modes is likely to represent noise in the original signal. Thus, by removing the higher modes a "cleaner profile" is obtained. Preisendorfer et al (1981) suggest that modes which can not be distinguished from randomly generated data should be removed. Dunteman (1989) suggests that all modes for which the eigenvalue is less than one should be removed. Dunteman's approach is used within this study.

C. DEVELOPMENT AND THEORY OF OBJECTIVE ANALYSIS

1. Development of Objective Analysis

Objective analysis is a technique that will produce an optimal estimate of some quantity at a given location by the interpolation of irregularly spaced data points. The method is based upon the Gauss - Markov theory.

Objective analysis was first used in meteorology by Gandin (1965) who used the technique to analyze atmospheric pressure and windfields. The technique was introduced for oceanographic use by Bretherton et al (1976), who demonstrated

its value in determining optimal temperature, velocity and streamline maps. The technique was applied by Freeland and Gould (1976) to data taken during POLYMODE and successfully produced stream function maps of the North West Atlantic.

Carter (1983) extended the use of objective analysis by considering distance variations separately in the X and Y directions and a temporal component, thereby allowing observations made at different places and at different times to be mapped. In addition, the theory allows an explicit statement to be made about the error in the determination of an interpolated value at a given location. Because of the introduction of a temporal component, Carter's method also enables maps of the quantity to be predicted for a future time.

Objective analysis is now widely used in oceanography. For instance, Watts et al(1989) used objective analysis to model the depth of the 12 degree Celsius isotherm from inverted echo sounder observations taken in the vicinity of the Gulf Stream. Objective analysis is a standard interpolation tool that is extensively used for computer aided numerical prediction in both meteorology and oceanography (see Clancy 1989).

2. Theory of Objective Analysis

The derivation outlined below, after Carter (1983), forms a statement of the Gauss Markov theory for determining a least squares optimal value.

The statistical model for objective analysis assumes a stationary h^mogeneous field. Let $\hat{\theta}_r$ be a measurement of some quantity and let the error in the measurement be e_r. Then,

$$\hat{\theta}_r = \theta_r + e_r$$

equation 17

where θ_r is the true value. It is assumed that observation error is uncorrelated with the true field such that,

$$R(e_r \theta_s) = 0$$

equation 18

where $R(e_r \theta_r)$ represents the correlation between the error e at position r and the measured field at some other locations.

It is also assumed that the correlation between observation errors at two locations is zero,

$$R(e_r e_s) = e^2 \delta_{rs}$$

equation 19

where $R(e_r e_s)$ represents the correlation between e_r and e_s, e^2 is the error variance, and δ_{rs} is the Krondiker delta having

a value of one when r equals s and the value of zero otherwise.

Objective analysis seeks to find the optimal value of a given quantity X at an arbitrary location. The optimal estimate of the value at the grid location is designated $\hat{X}$. In matrix form the estimate at the grid points is given as a linear combination of the values of the data measured at a variety of locations r such that,

$$\hat{\boldsymbol{X}}=\boldsymbol{A}\boldsymbol{\theta}_r$$

equation 20

where $\boldsymbol{\theta}_r$ is the value of the quantity measured at position r throughout the region. For example $\boldsymbol{\theta}_r$ could represent sea surface temperature measurements taken at various irregularly spaced positions within a given region. Whereas $\boldsymbol{X}$ represents true values, the value $\hat{\boldsymbol{X}}$ is the estimate that is determined at the grid points by interpolating the values of $\boldsymbol{\theta}$ onto the grid by the use of linear combinations of $\boldsymbol{\theta}_r$ using the matrix $\boldsymbol{A}$.

In order to determine the estimates at the chosen grid points it is first necessary to ascertain values for the elements in matrix $\boldsymbol{A}$. This is done in such a manner as to give the optimal estimate of $\hat{\boldsymbol{X}}$. Throughout the derivation $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$

are referred to as if they were known, whereas in fact they are the quantities ultimately that are to be determined.

Initially it is the value of $\boldsymbol{A}$ that is sought such that it minimizes the error by a least squares fit between the true value of $\boldsymbol{X}$ at the grid points and the estimate $\hat{\boldsymbol{X}}$. Firstly let

$$C_{x\theta}=E[x\theta']$$

equation 21

where $C_{x\theta}$ is the correlation matrix found by comparing the value of the quantity at the required grid point locations compared with those at the given data sites.

$$C_{x}=E[XX']$$

equation 22

where C_x is the correlation between the value at any required grid point location compared to the value at any other required grid point location.

$$C_{\theta}=E[\theta\theta']$$

equation 23

where C_θ is the correlation between the values at any two data point sites.

Then to obtain the optimal interpolation the value of the error $\boldsymbol{C_e}$ is minimized such that

$$\boldsymbol{C_e}=E[ee']=E[(\hat{X}-X)(\hat{X}-X)']$$

equation 24

where $\boldsymbol{C_e}$ represents the correlation between the mean square variance of the estimated values compared to the actual values. Substituting equation 20 into equation 23 and expanding gives,

$$\boldsymbol{C_e}=E[(\boldsymbol{A\theta}-\hat{\boldsymbol{\theta}})(\boldsymbol{A\theta}-\hat{\boldsymbol{\theta}})']$$

equation 25

and,

$$\boldsymbol{C_e}=\boldsymbol{AC_\theta A'}-\boldsymbol{C_{x\theta}A'}-\boldsymbol{AC_{\theta x}}+\boldsymbol{C_x}.$$

equation 26

This expression can be simplified by using a matrix identity and noting that $\boldsymbol{C_{\theta x}}=\boldsymbol{C'_{x\theta}}$,

$$\boldsymbol{C_e}=(\boldsymbol{A}-\boldsymbol{C_{x\theta}C_\theta^{-1}})\boldsymbol{C_\theta}(\boldsymbol{A}-\boldsymbol{C_{x\theta}C_\theta^{-1}})'-\boldsymbol{C_{x\theta}C_\theta^{-1}C_{x\theta}}+\boldsymbol{C_x}.$$

equation 27

Since the matrices $\boldsymbol{C_\theta}$ and $\boldsymbol{C_\theta^{-1}}$ are nonnegative definite, then the error matrix is minimized when,

$$\boldsymbol{A}-\boldsymbol{C_{x\theta}}\boldsymbol{C_\theta^{-1}}=0$$

equation 28

giving,

$$\boldsymbol{A}=\boldsymbol{C_{x\theta}}\boldsymbol{C_\theta^{-1}}.$$

equation 29

The value of the error matrix $\boldsymbol{C_e}$ can be written explicitly as,

$$\boldsymbol{C_e}=\boldsymbol{C_x}-\boldsymbol{C_{x\theta}}\boldsymbol{C_\theta^{-1}}\boldsymbol{C'_{x\theta}}.$$

equation 30

From equation 29 and substituting for $\boldsymbol{A}$ in equation 20, the estimate of the value of the quantity at the grid points is given by,

$$\hat{X}=C_{x\theta}C_{\theta}^{-1}\theta_{r}$$

equation 31

and the error in these estimates is given by equation 30.

Thus, providing the correlation matrices $C_{x\theta}$ and C_{θ} can be determined a value for $\hat{X}$, the estimate of the value at any given grid location can be obtained from a knowledge of θ, the value at any given location. Equations 29 and 30 are a statement of the Gauss Markov theory.

3. The correlation function

The correlation matrix $C_{x\theta}$, a measure of the correlation between the values at each of the data sites compared to the values at each of the grid points, is unknown. Similarly, the correlation matrix C_{x}, the correlation between successive grid point values, is also unknown. The only correlation that is available is C_{θ}, the correlation between data values at the irregularly sampled locations.

However, the determination of C_{θ} is not straight forward. In order to determine the correlation between two points it is necessary to have made several readings at each location, whereas in this study only one reading at a given location is available. This problem is overcome by assuming that the correlation between any two points is a function of distance.

The correlation matrix C_θ is formed by computing the distance between each data point and every other data point. The data pairs are grouped into distance bins, and the correlation between distance bins is then determined using the expression,

$$\theta_{rs}=\frac{\sum(\theta_r-\overline{\theta_k})(\theta_s-\overline{\theta_k})}{(\sum(\theta_r-\overline{\theta_k})^2\sum(\theta_s-\overline{\theta_k})^2)^{1/2}}$$

equation 32

where θ_r and θ_s are data values at two points r and s, and $\overline{\theta_k}$ is the mean of the values for distance bin k. Once the correlation function has been determined for the data points within the region the results are applied to the two unknown matrices $C_{x\theta}$ and C_x. Simply knowing the distance between a grid point and a data point or between two particular grid points is sufficient information to enable the corresponding correlation between the two points to be computed. Unfortunately there is one more slight complication, in that the two matrices have to be, by definition, positive definite for equation 31 to be valid. This means that an estimate of the correlation between two successive points can not be achieved from a database simply by interpolating between two adjacent distance bins, because the approximation may not be positive definite. In order to ensure that the two matrices

are positive definite it is necessary to fit a function to the distance correlation database.

The function that is normally fitted to the curve (Carter and Robinson 1987) takes the form,

$$C_{rs}=(1-(r/a)^2)\,e^{-(r/b)^2}$$

equation 33

where a and b are the unknowns to be determined, r the distance between any two data points r and s, and C_{rs} the correlation between them. The values of a and b are determined iteratively by minimizing the error between the original correlations C_n as given in the database outlined above and C_{rs}. where the error is given by,

$$\varepsilon=(C_{rs}-C_n)^2.$$

equation 34

The correlation matrices of $\boldsymbol{C_{x\theta}}$ and $\boldsymbol{C_x}$ can now be determined from the function described in equation 33 and the objective analysis can be undertaken.

D. INTERPOLATED ERROR

Having obtained an estimate for the value of the first four modal amplitudes at each of the grid positions, it then remains to use the theory of EOFs to reconstruct a synthetic XBT at each of these positions. This is simply achieved by multiplying each modal amplitude by its corresponding eigenvector and adding the four resulting vectors together, as per equation 16. However some of the estimated modal amplitudes used contain error. The error variance of each modal amplitude is specified by equation 30, and must be taken into account in reconstructing a synthetic XBT at a grid position.

Consider a modal amplitude at a particular grid location Q_i having an error variance ϵ_Q^2, and assuming the synthetic XBT at that position is going to be reconstructed using i EOFs, then the error variance in the synthetic XBT ϵ_θ^2 can be shown (Carter 1983) to be given by,

$$\epsilon_\theta^2=\sum_i Y_i^2\epsilon_{Qi}^2 .$$

equation 35

The error variance from this reconstruction is then mapped to give a pictorial image of areas within the region that have high and low error variances.

The error variance is a measure of the confidence of a given reconstruction. Figure 2 shows an example of an error variance map. Low confidence is indicated when the values approach one. This map is the combination of the individual modal amplitude error maps shown in Figure 3 using equation 35. The figure also shows where each XBT cast was taken. As would be expected, the lowest error variance (highest confidence) occur in areas that have a high number of samples, with the error variance (lowest confidence) being largest where there are no or few samples.

E. APPLICATION OF THEORY TO CURRENT STUDY

All the elements of the theory can now be put together to analyze the area under investigation. Firstly the original XBTs will be converted into a correlation matrix, where one depth is compared to another and the whole data set ensemble averaged to give the projection matrix. This matrix will then be decomposed to find the significant eigenvectors, noting the value of the corresponding eigenvalues. The most significant eigenvectors or modes will be selected, and for each XBT within the set the corresponding modal amplitudes will be determined.

Once the modal amplitudes have been found, a correlation matrix as a function of distance can be constructed. From this an appropriate function will be fitted and the correlation

matrices C_x and $C_{x\theta}$ determined. The modal amplitudes can then be optimally interpolated onto a grid. The process is repeated for the second, third, and fourth modal amplitudes.

Synthetic XBTs can then be reconstructed at each grid point using the interpolated modal amplitudes and a measure of the error variance in each XBT can be determined from the error matrices generated by the objective analysis.

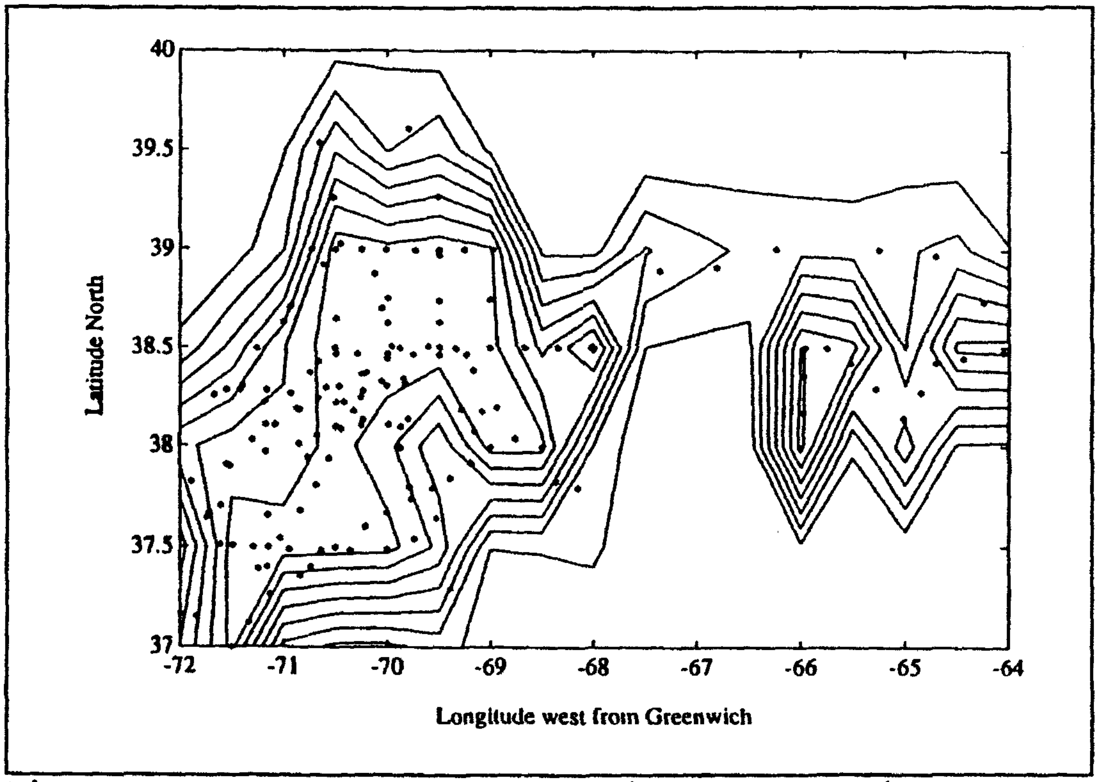

Figure 2 Reconstructed error variance map using all 156 XBT's. The map is produced using equation 37 (effectively combining the four maps from Figure 2. The contours are at 0.1 spacing. The central contour represents 0.1 (or 10%) error variance.

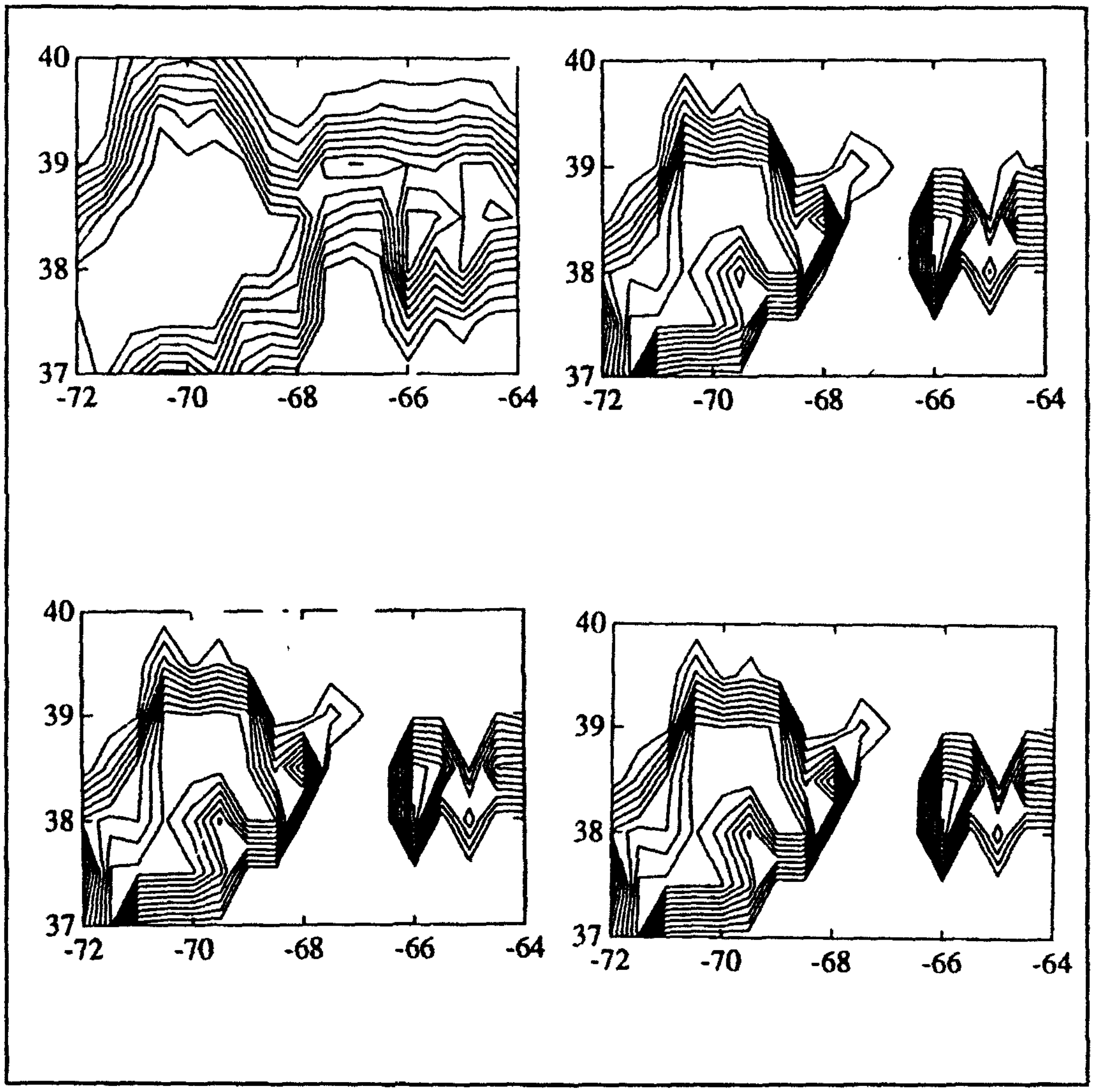

Figure 3 Error variance maps for the first four modal amplitudes. Top left shows error variance for first, top right for second, bottom left and right show third and fourth respectively. At 0.1 intervals. Inner most contour representing 0.1 or 10% error variance.

III DATA

A. THE MEANDER EXPERIMENT

The data for this study consists of 156 XBTs taken within the region of a Gulf stream meander sampled during the period September 17th to October 13th 1988. The original data was collected as part of a much wider experiment that involved two cruises, one in the autumn of 1988 and the second in the spring of 1989. The first cruise samplea an anticyclonic meander crest (EN 185) where the path of the current is convex to the North. The second cruise collected data from a cyclonic meander trough (EN 194) where the flow of the current is convex to the South. The objective of the two cruises was to investigate the time dependent kinematics and dynamical structures of Gulf Stream meanders. The Gulf Stream meander was sampled with a variety of instruments, and density and velocity fields were computed to enable fluxes of mass, momentum and vorticity to be determined as the meander progressed in space and time.

B. THE XBTs

The following technical details of the XBTs are taken from the initial cruise report (Hummon 1991).

The XBTs used in the survey were Sippican T7 probes which have a nominal depth rating of 760 metres. The XBTs were launched from a fixed stern deck launcher with a BathySystemn 810 XBT deck unit. The data was stored on a HP-85B computer equipped with an HP9121D disk drive. The software was supplied by BathySystems but was substantially modified to allow simpler and faster processing. The raw data was recorded in volts versus descent time, The data were transferred to a MassComp computer and each profile was converted into temperature versus depth measurements and stored onto disk or magnetic tape.

The resolution of the data is 0.65 metres with a 0.1 metre precision. The stated accuracy of the depth measurement is five metres or 2% of the depth, whichever is greater. Temperature data is stored to within 0.001 degree Celsius with measurement accuracy to within 0.15 degrees Celsius.

The data was edited to remove the first three measurements corresponding to depths less than two metres. Readings taken at depths greater than 810 metres, outside of the stated operating range of the probes, were also removed. Spikes, bad data and wire breaks in individual profiles were deleted by hand on the MassComp computer. The full set of XBT casts is shown in Figures 4-17. The geographic distribution of the casts is shown in Figure 2, with location values being given in the log shown in Appendix A.

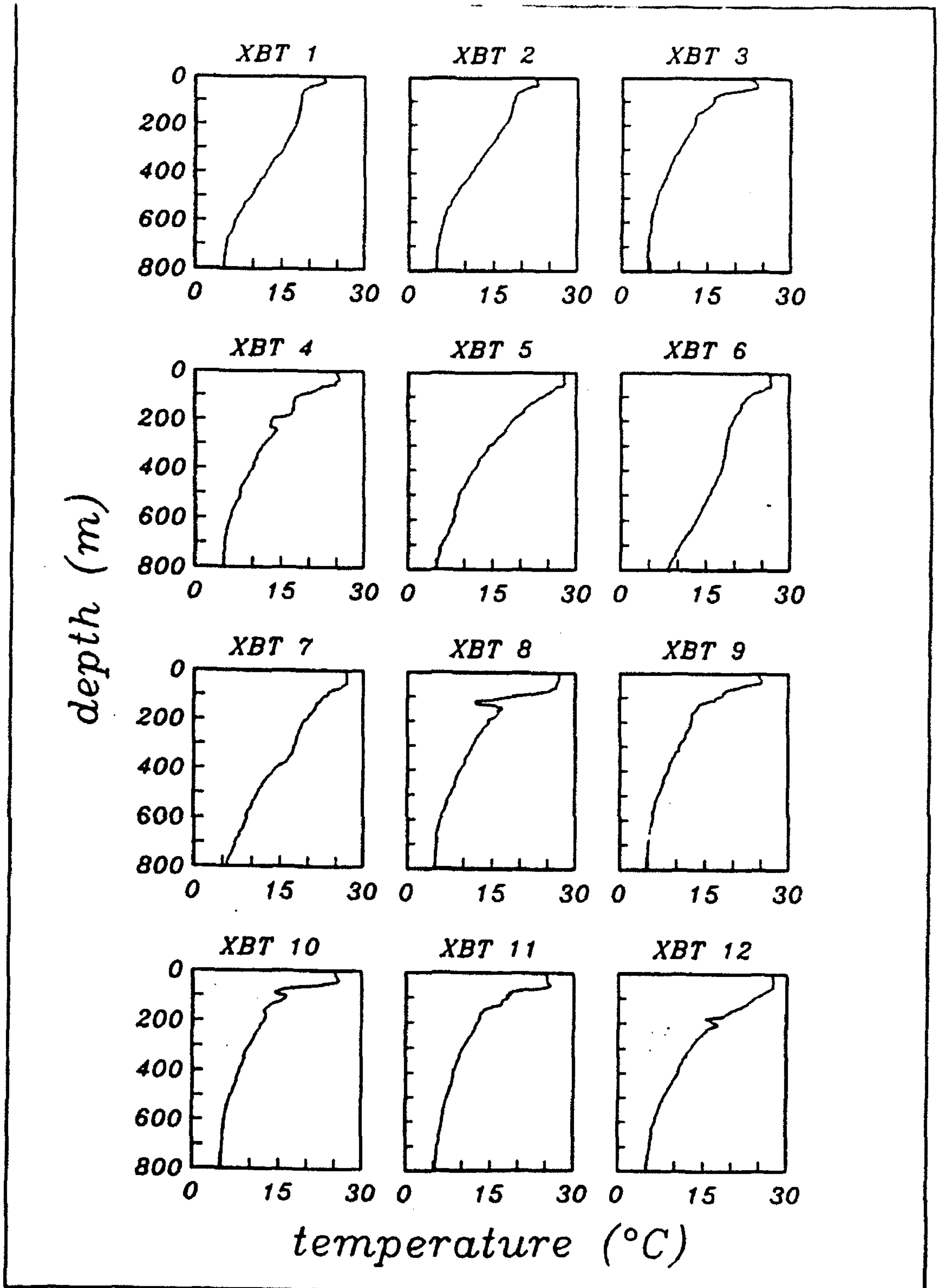

Figure 4 - 17 show all the XBT casts taken during the Anatomy of a Meander experiment.

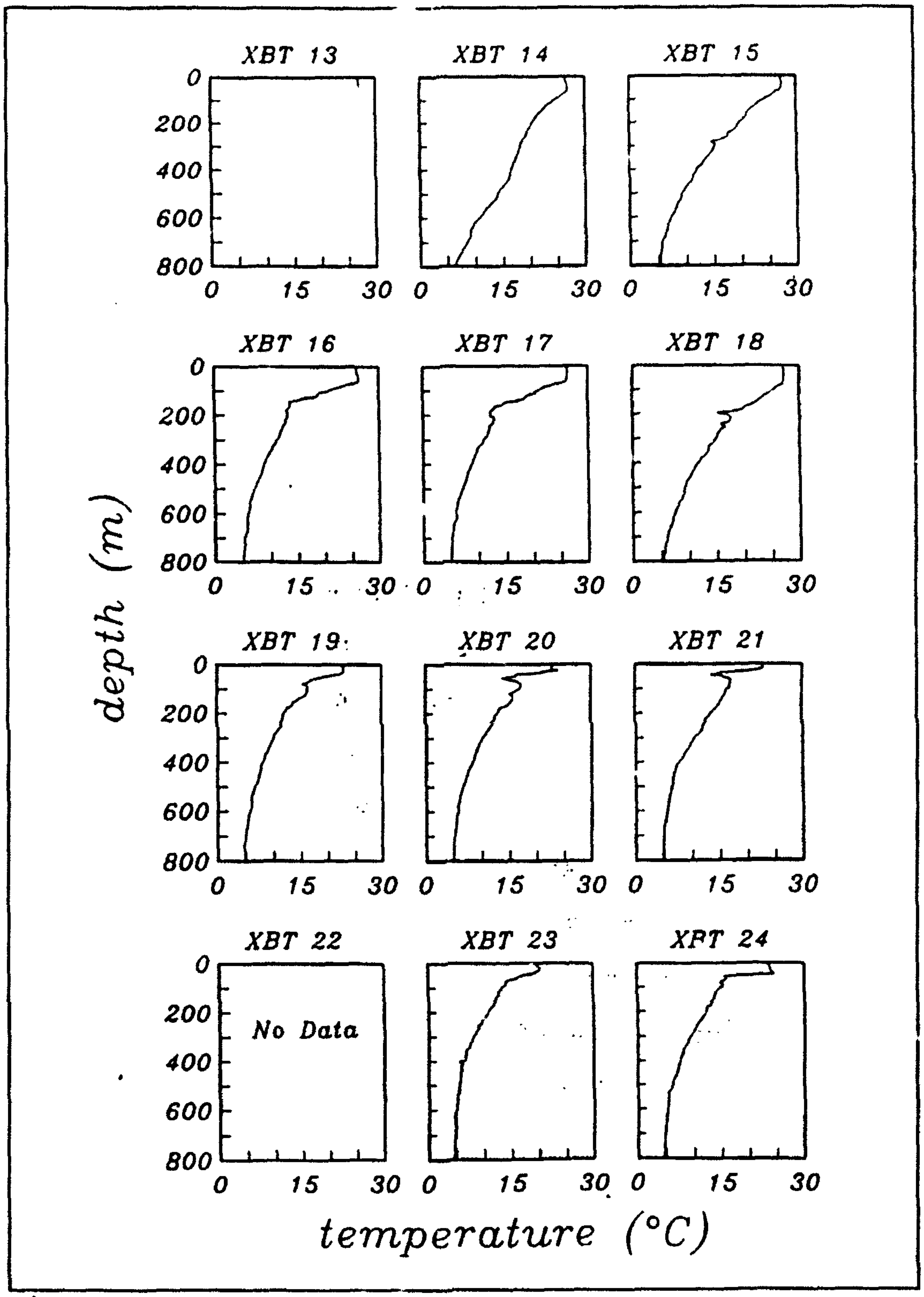

Figure 5 XBTs 13 - 24

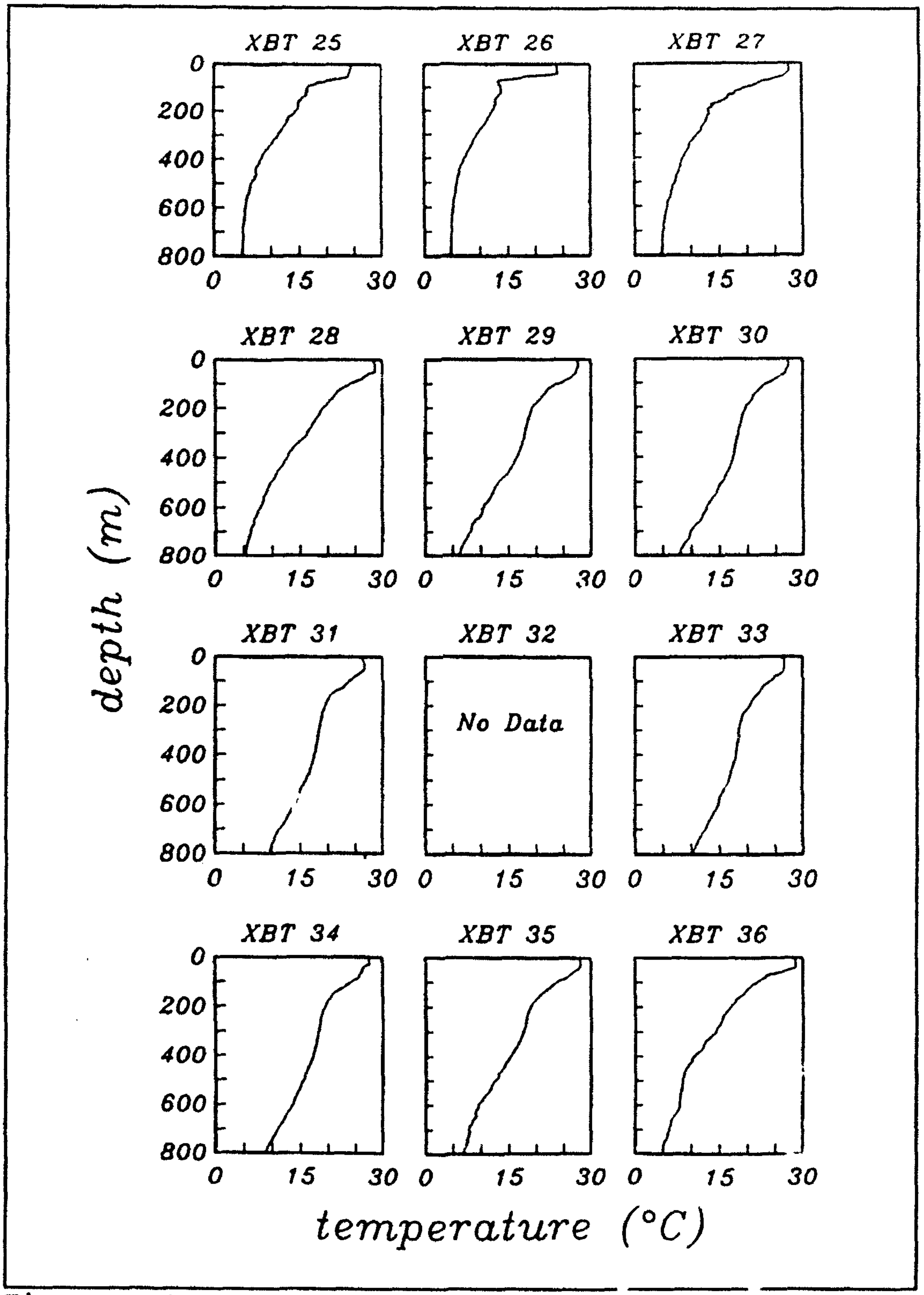

Figure 6 XBTs 25 - 36

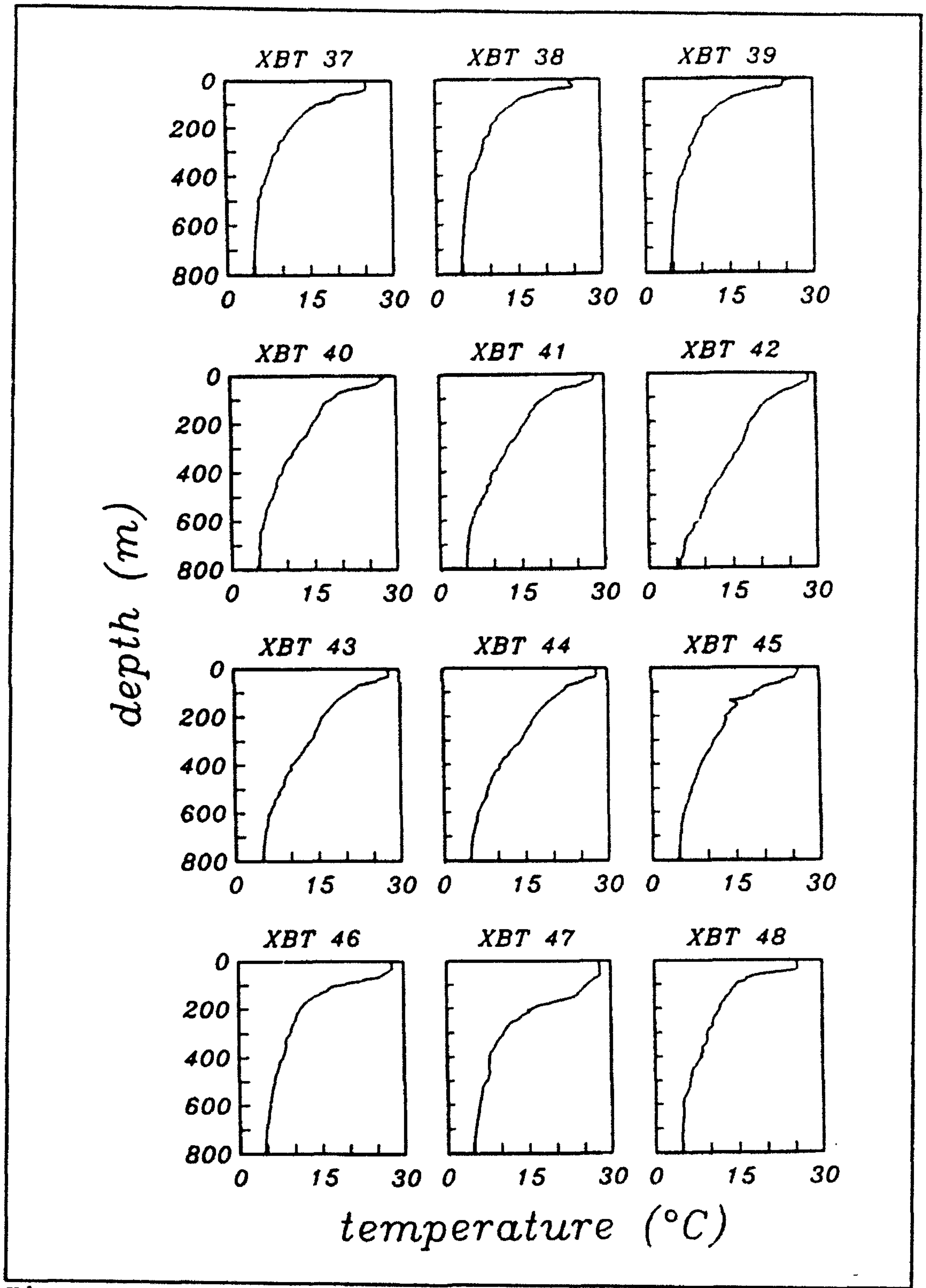

Figure 7 XBTs 37 - 48

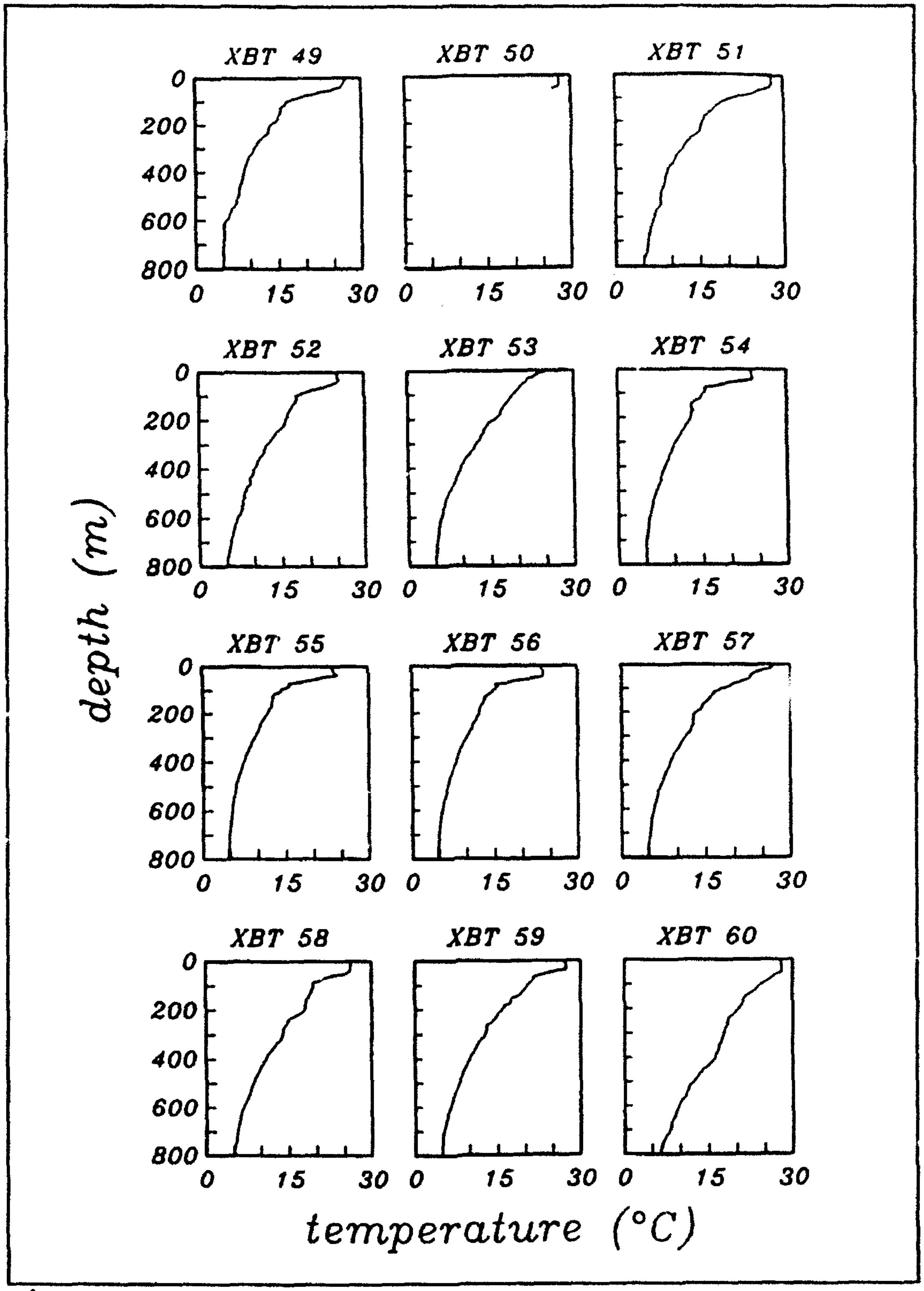

Figure 8 XBTs 49 -60

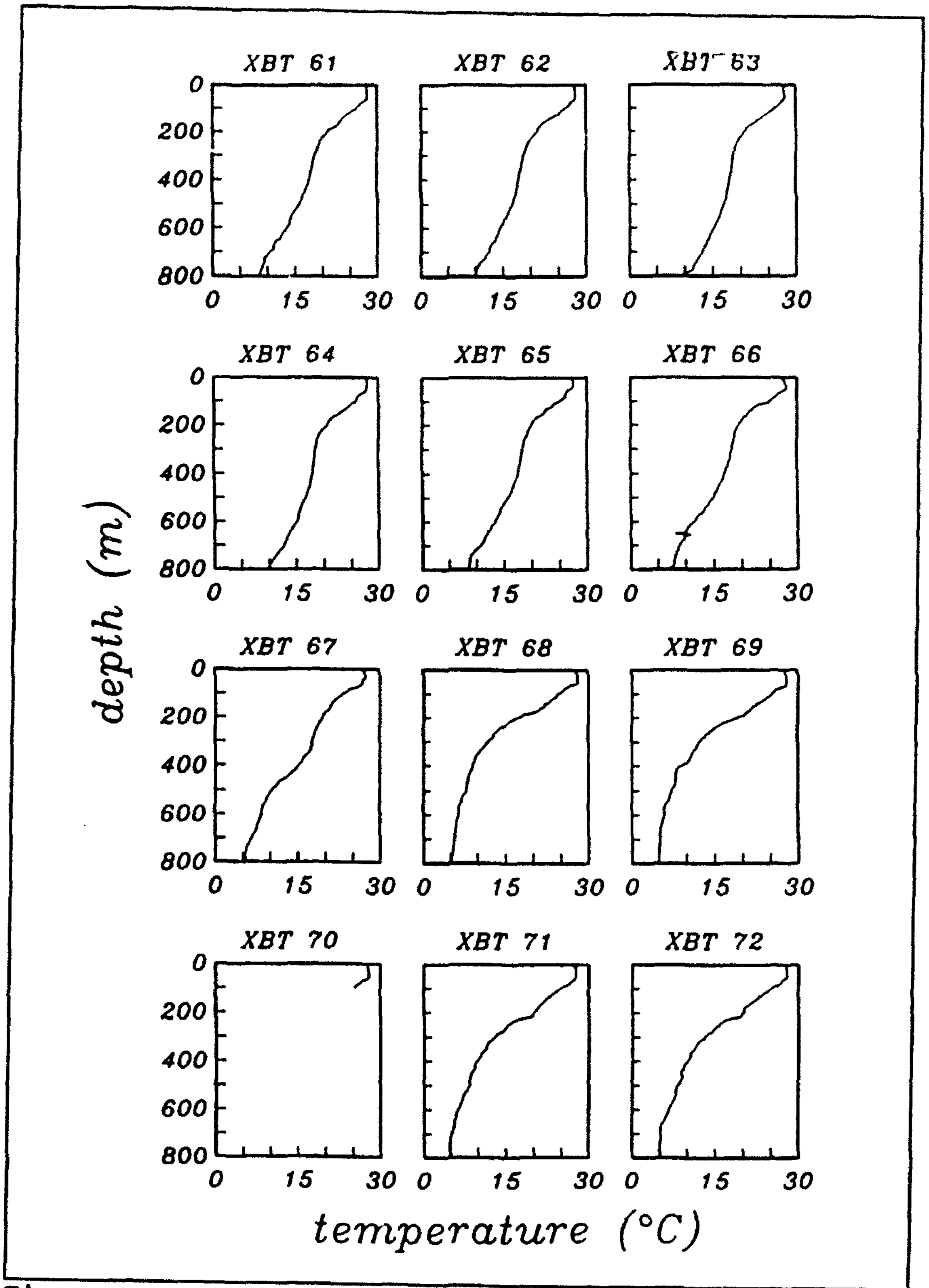

Figure 9 XBTs 61-72

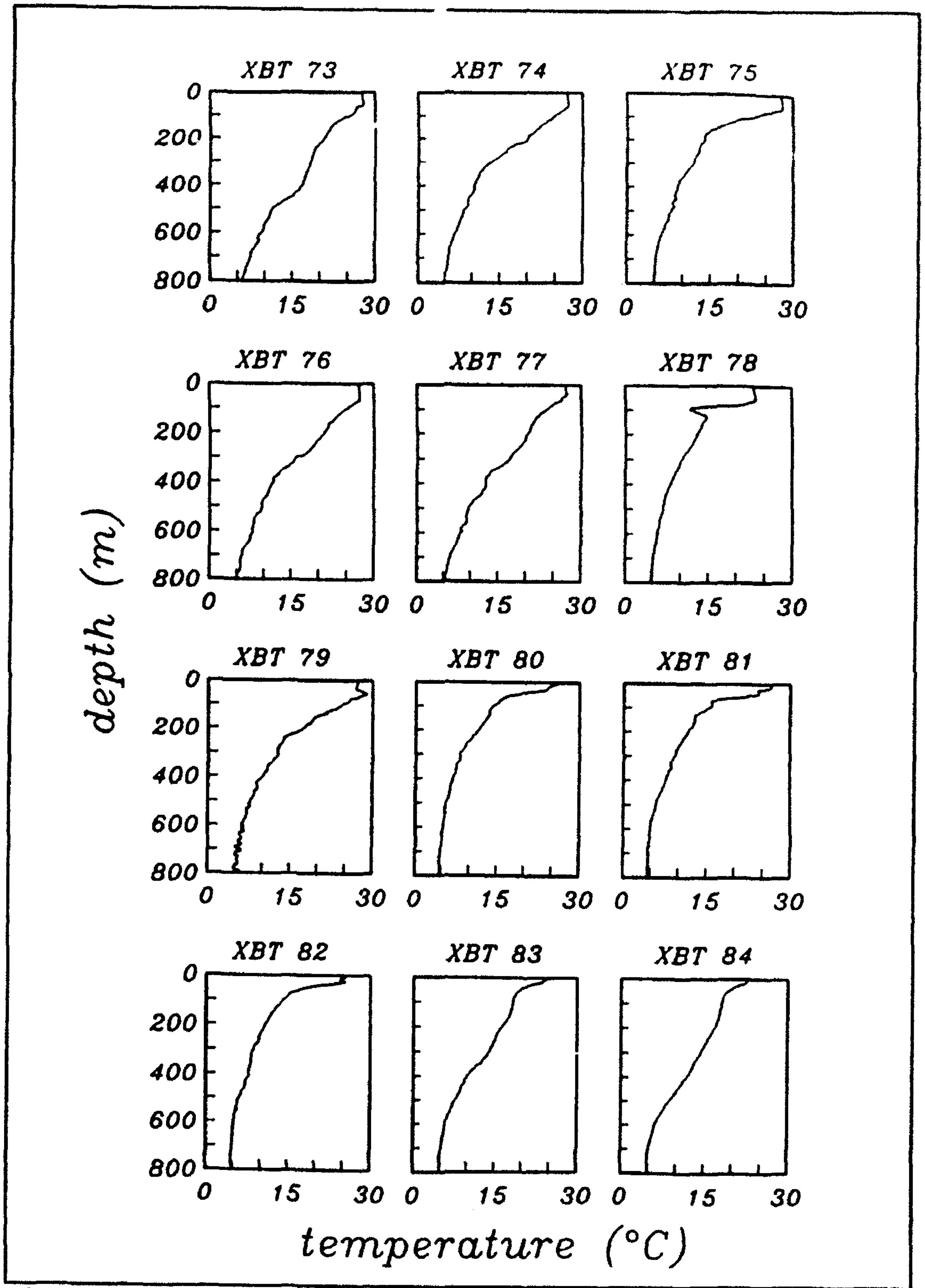

Figure 10 XBTs 73 -84

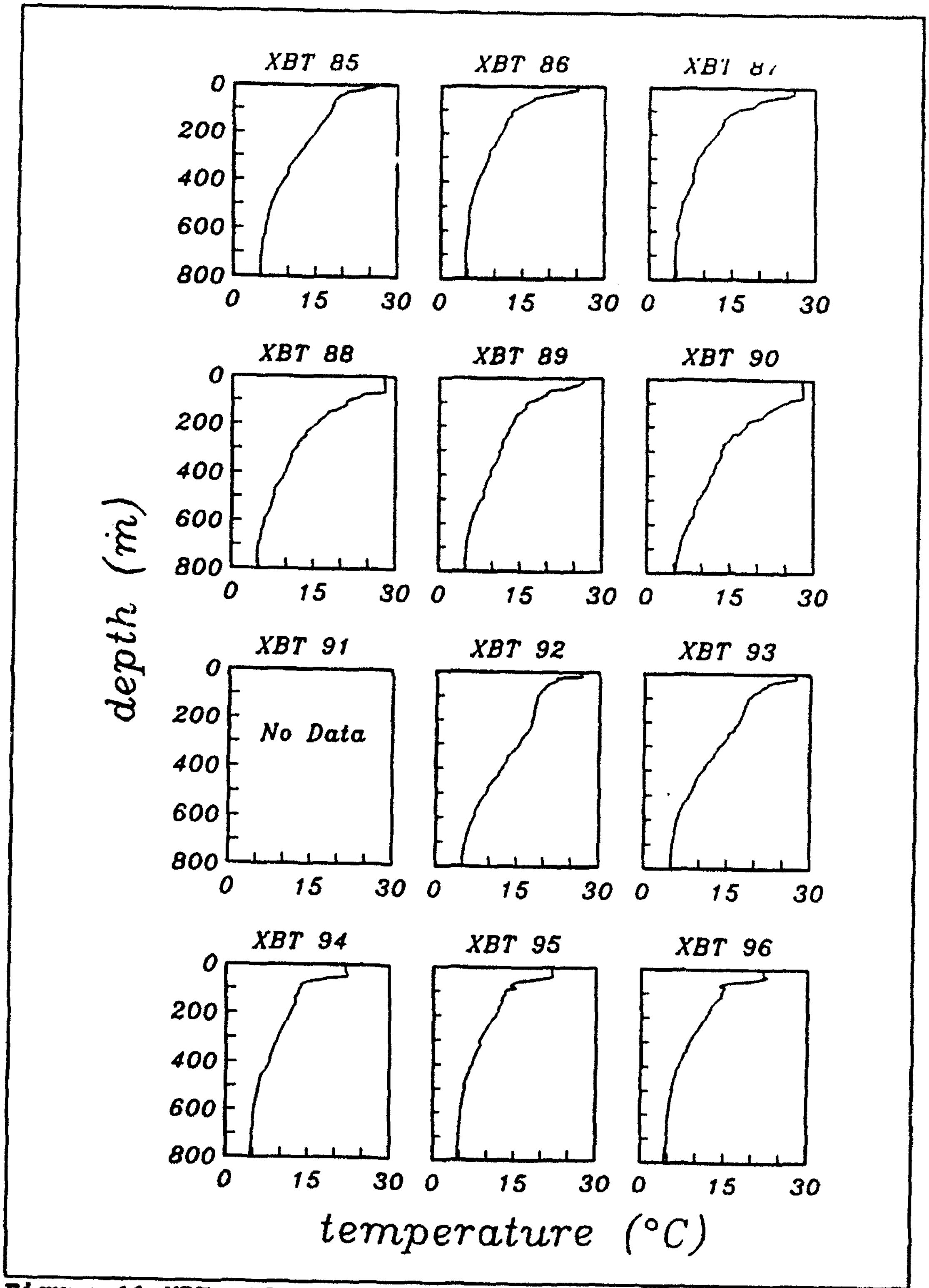

Figure 11 XBTs 85-96

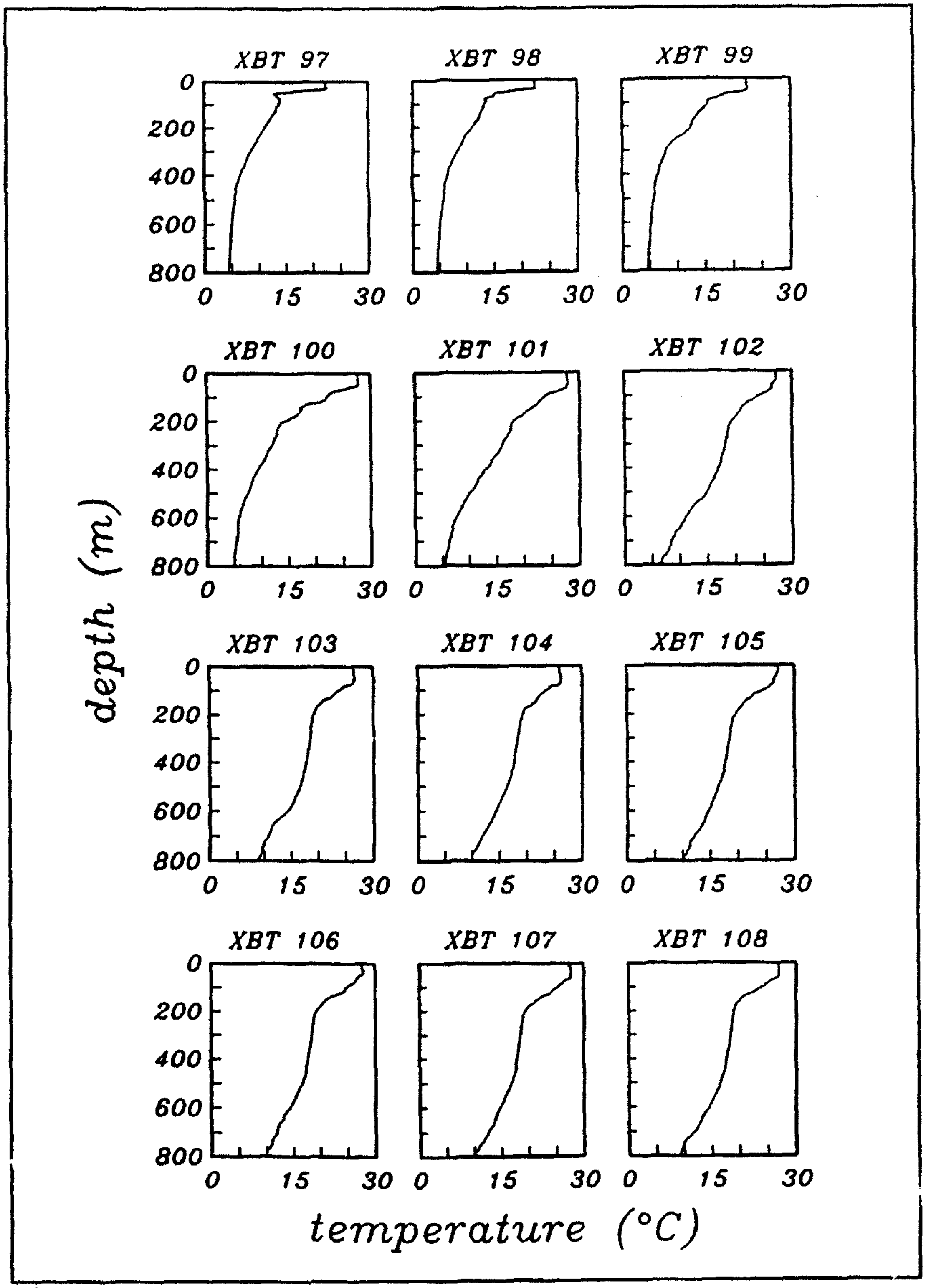

Figure 12 XBTs 97 - 108

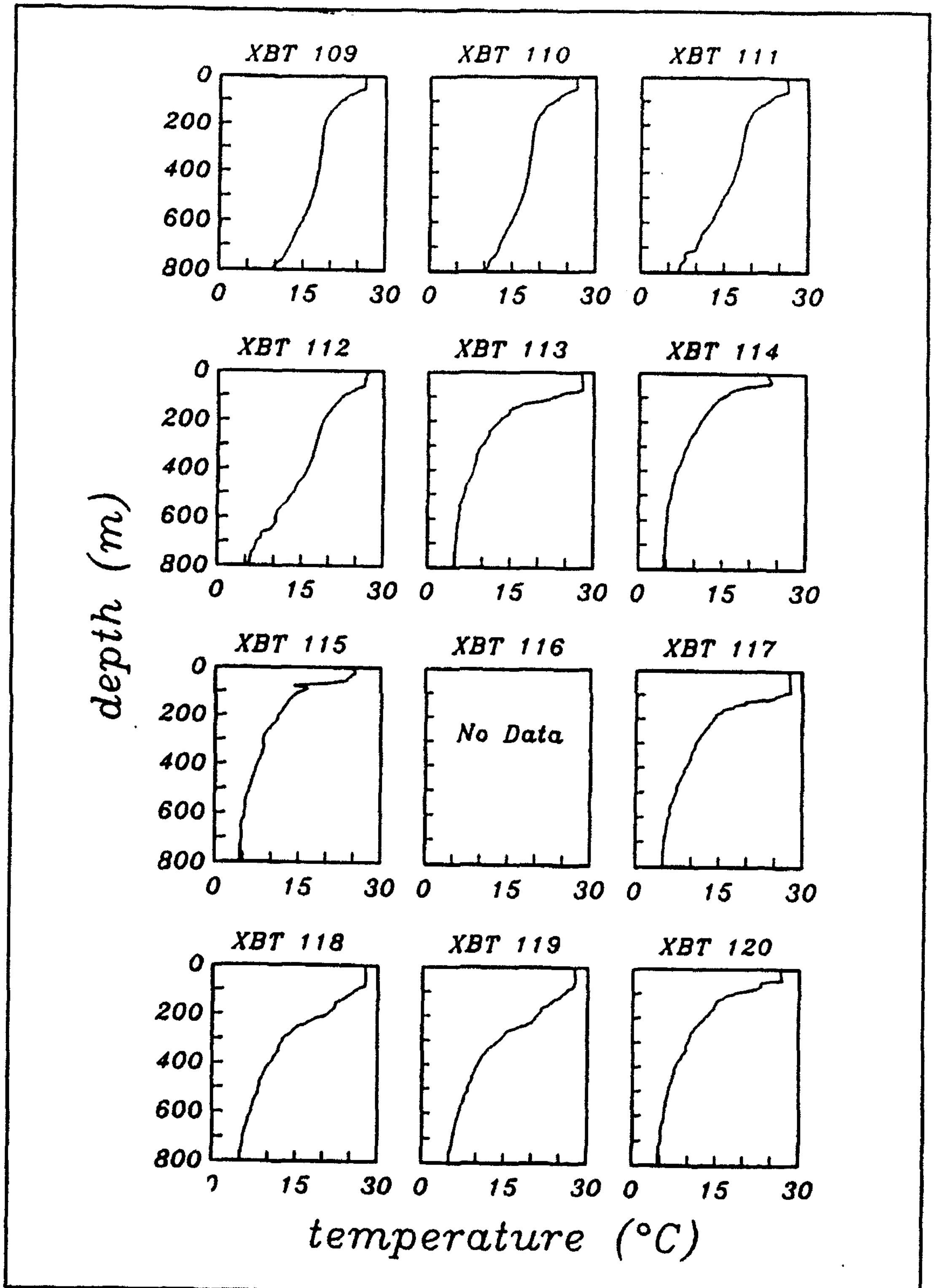

Figure 13 XBTs 109 - 120

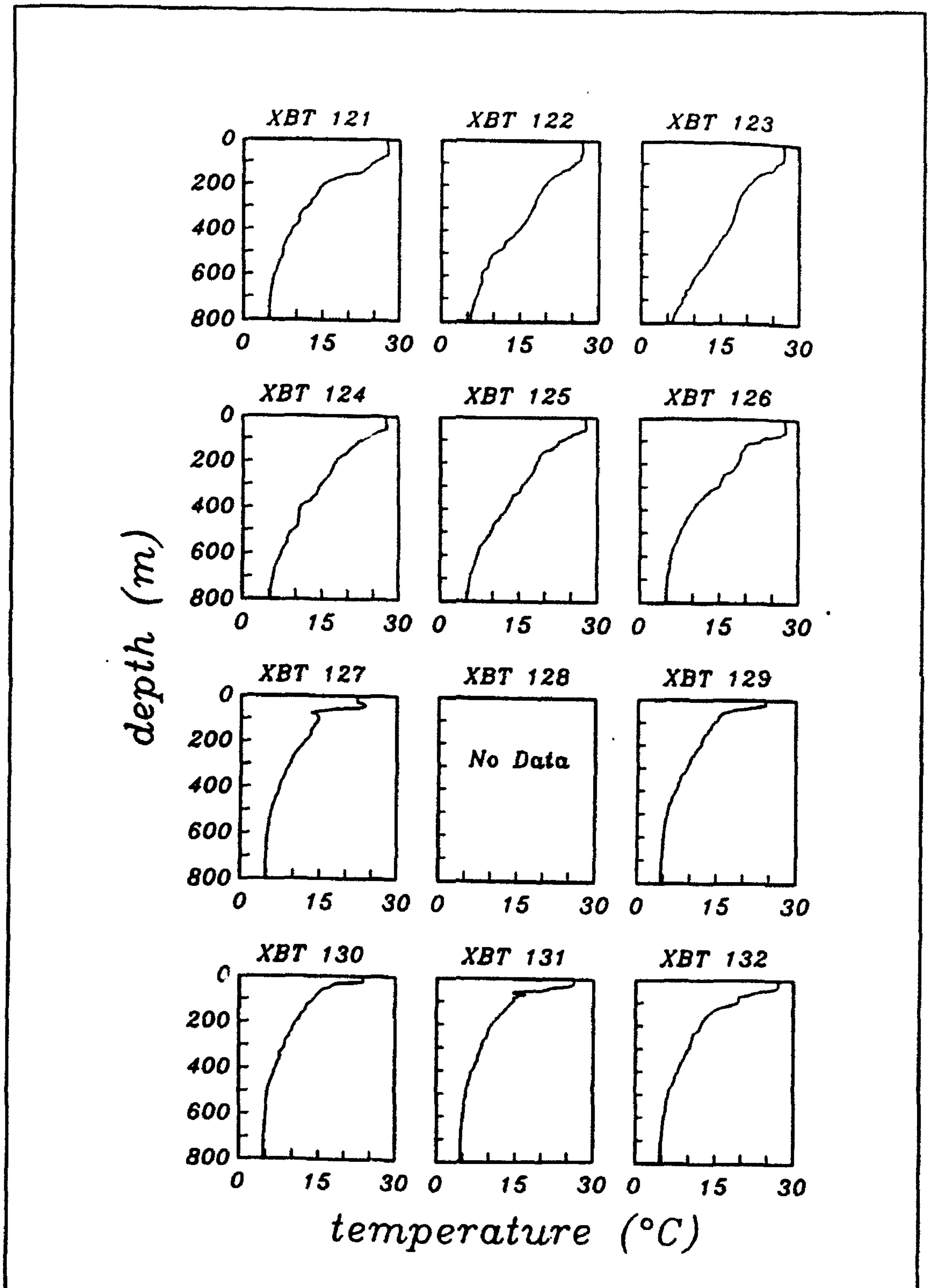

Figure 14 XBTs 121 - 132

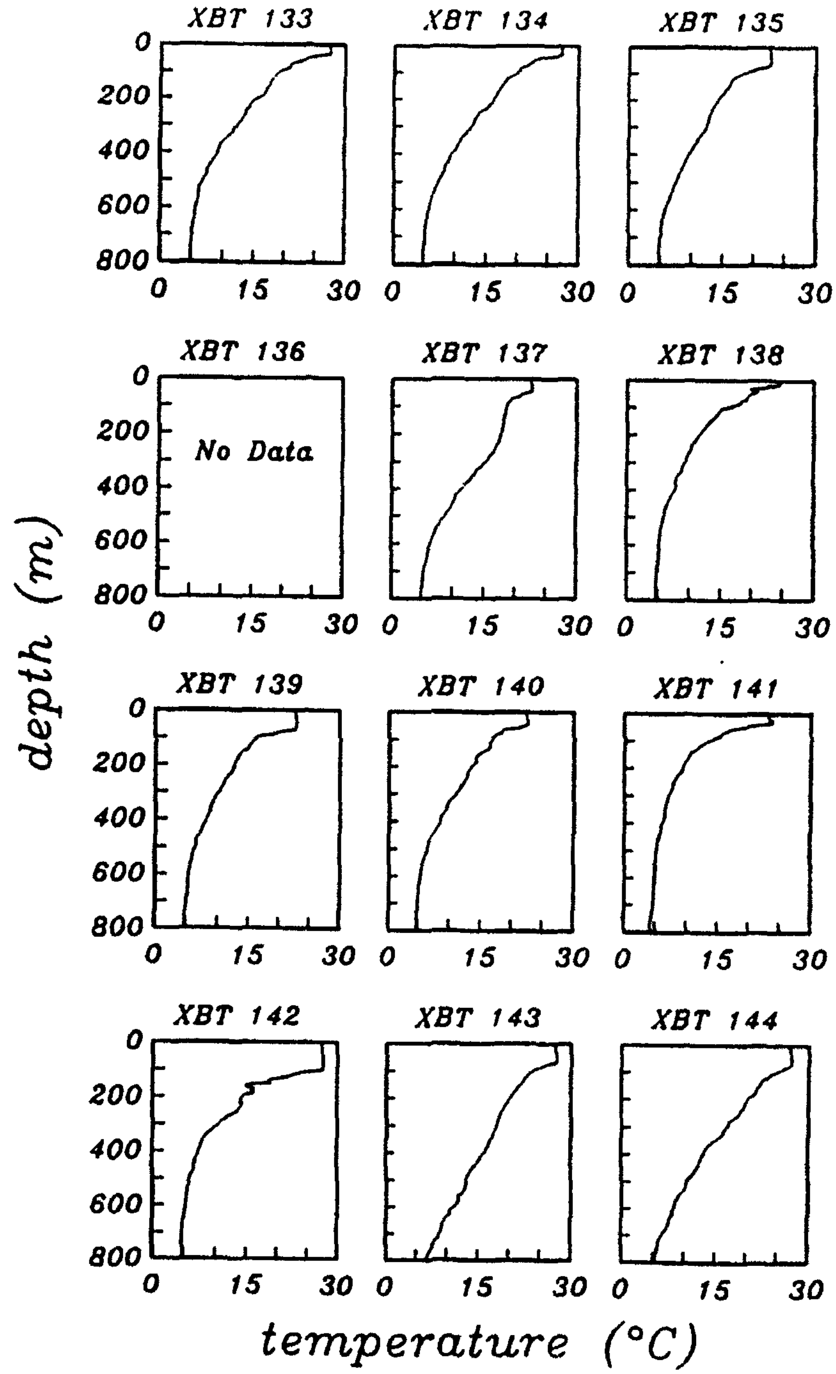

Figure 15 XBTs 133 - 144

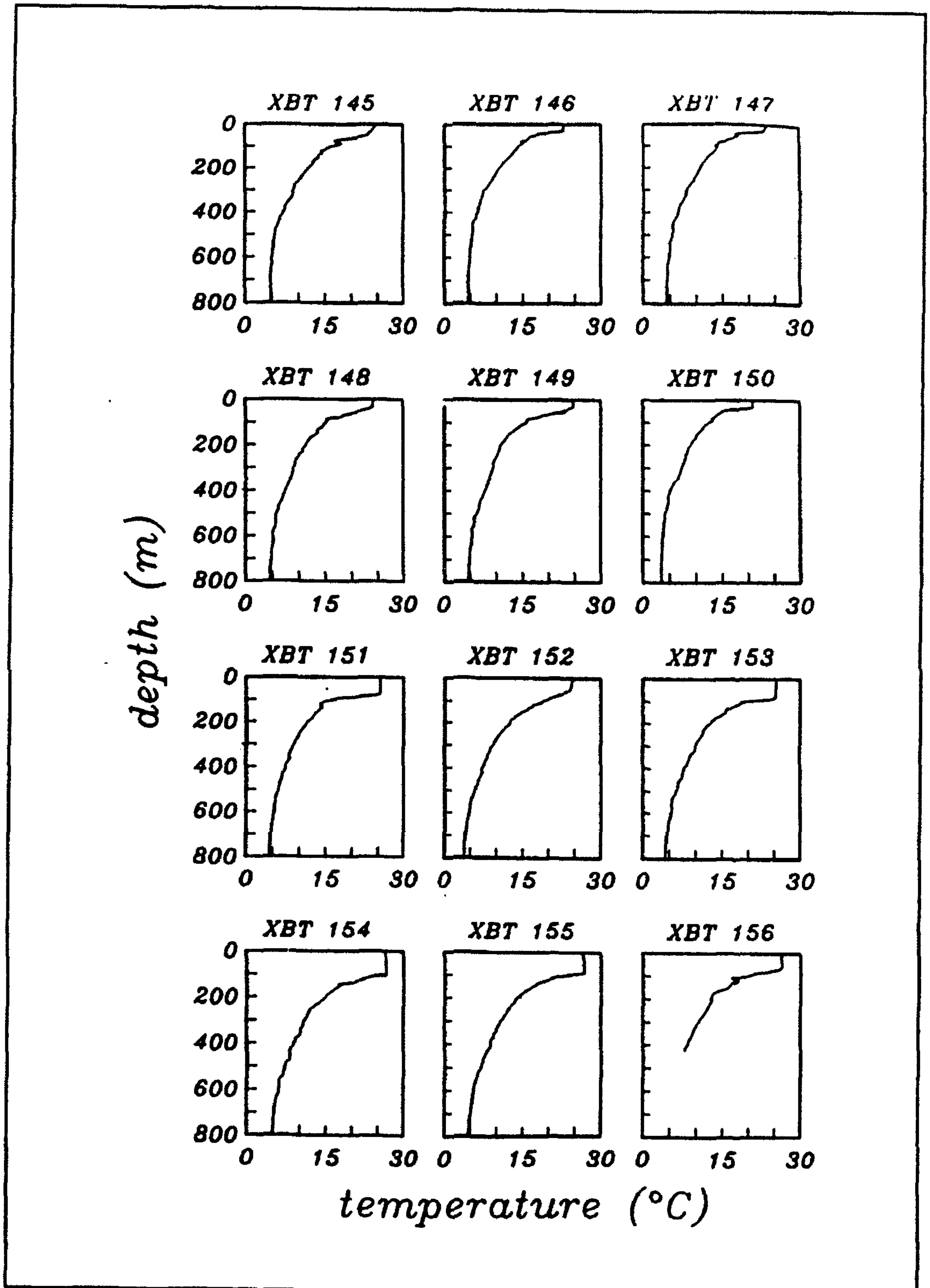

Figure 16 XBTs 145 - 156

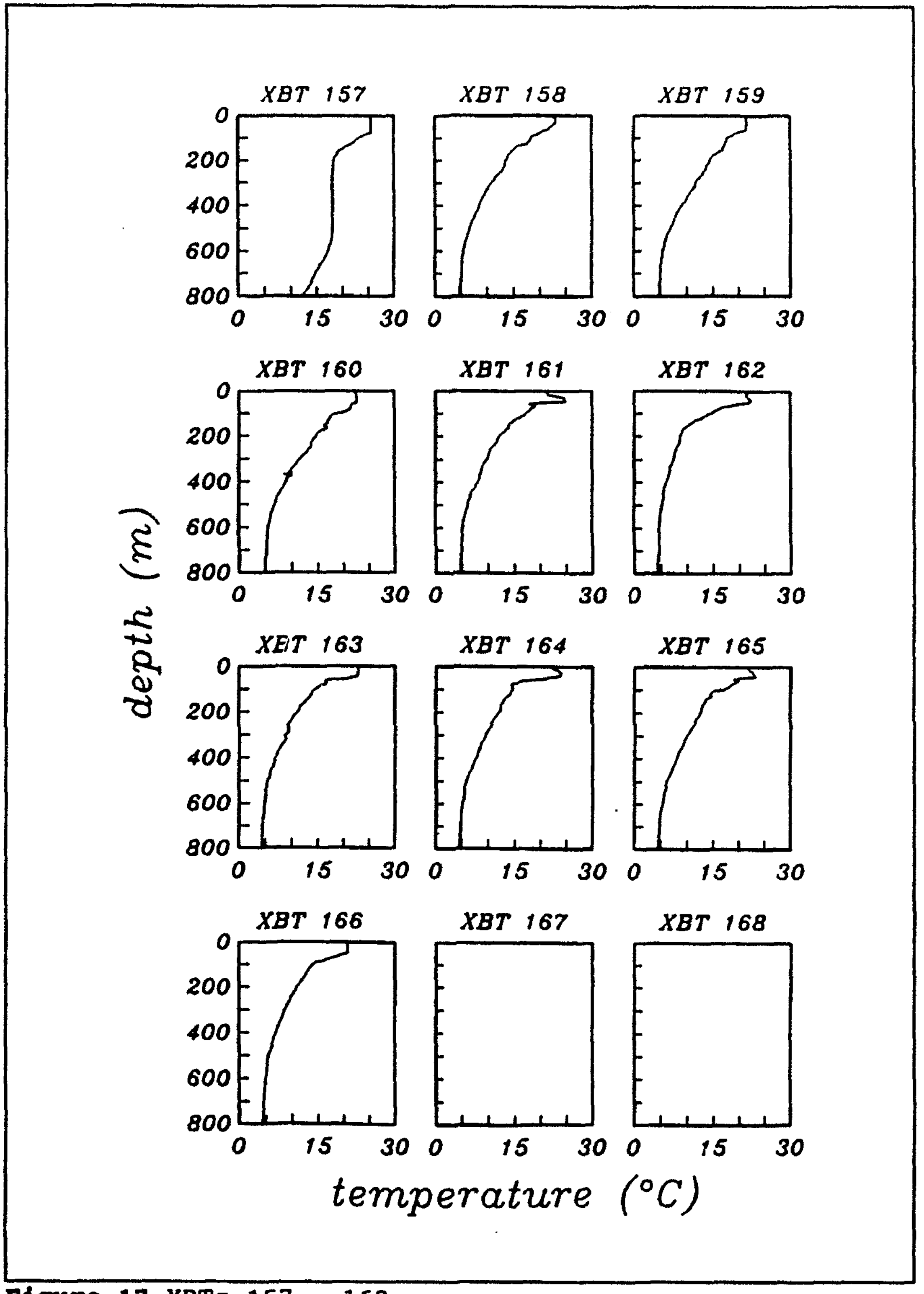

Figure 17 XBTs 157 - 168

IV METHODS

Part A of this chapter describes how the original projection matrix was computed and how the eigenvectors and eigenvalues were determined. Part B describes how the Objective Analysis was implemented and how the synthetic XBTs were reconstructed. Finally part C describes how the data set was reduced to find the minimum number of XBT sites that were required in the case of this particular Gulf Stream meander.

A. DEPTH CORRELATION MATRIX

1. The matrix

To overcome initial data analysis problems all XBTs less than 800 metres were removed from the data set. This left a total of 156 useable XBTs for further analysis.

The vertical correlation matrix was formed using FORTRAN program LOADBATHYS (Appendix 1B) and subroutine REDATA (Appendix 2B). The subroutine interpolates temperature values from each XBT at 10 metres intervals commencing with a depth of five metres. The vertical correlation matrix was computed in the main program by comparing the temperature at one depth with that at another depth. This process ensemble averaged over all 156 XBTs using equation 36.

$$A = \sum (\theta_i - \bar{\theta}_i)(\theta_j - \bar{\theta}_j) / (\sum (\theta_i - \bar{\theta}_i)^2 (\theta_j - \bar{\theta}_j)^2)^{1/2}$$

equation 36

where A is the 80 by 80 projection matrix formed by comparing the temperature at all 80 depths with each other and θ is the temperature at depths i and j, with the overbar representing the mean temperature for that depth i or j. The projection matrix is visualized in Figure 18.

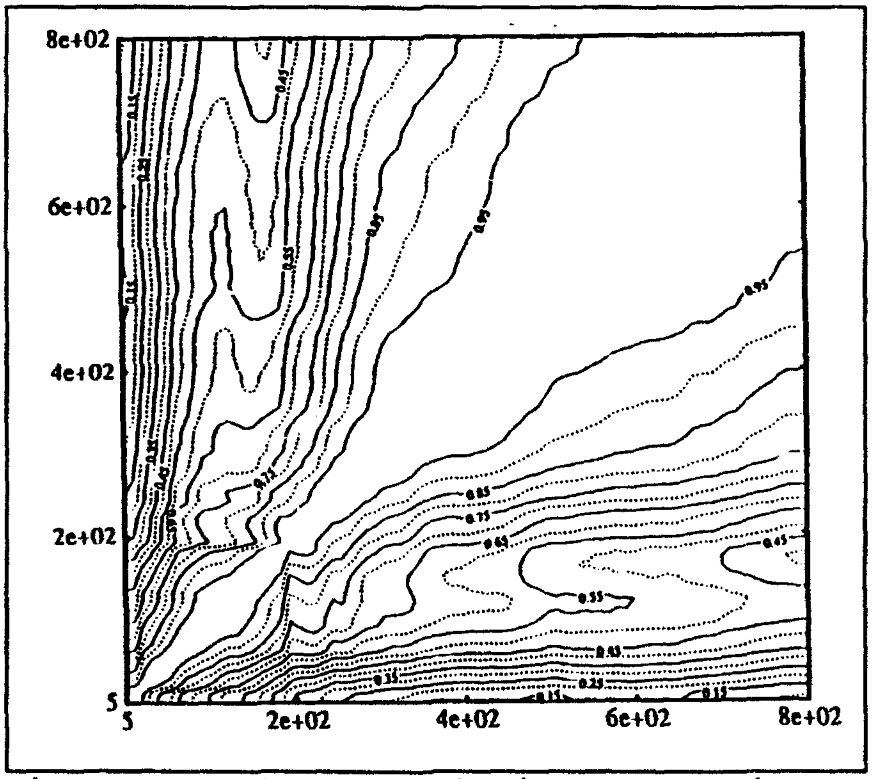

Figure 18 Contour map showing correlation of temperature between depths.

2. Eigenvectors and values

The correlation matrix, $\mathbf{A}$, was decomposed to find its eigenvalues and eigenvectors using equation 9.

Figure 19 shows the first six eigenvalues and the associated variance for the first four modes. Each eigenvalue is proportional to the variance contributed by its corresponding eigenvector (Harman 1976). The first eigenvector accounts for over 75% of the variance of the correlation matrix, the second eigenvector is responsible for 15%, the third for 5.1%,and the fourth for 1.7%. The cumulative percent variance explained by the first four eigenvectors is over 98% of the total variance of the projection (correlation) matrix. Thus, instead of using 80 eigenvectors to describe the variance in the correlation matrix $\mathbf{A}$, it is possible, using the criteria discussed by Dunteman (1989), to describe the matrix sufficiently with only four, with a minimal loss in information, thereby saving considerably on data storage and processing requirements and suppressing the noise contained within the higher modes. The modal amplitudes for each XBT were calculated using equation 16 in a MATLAB subroutine.

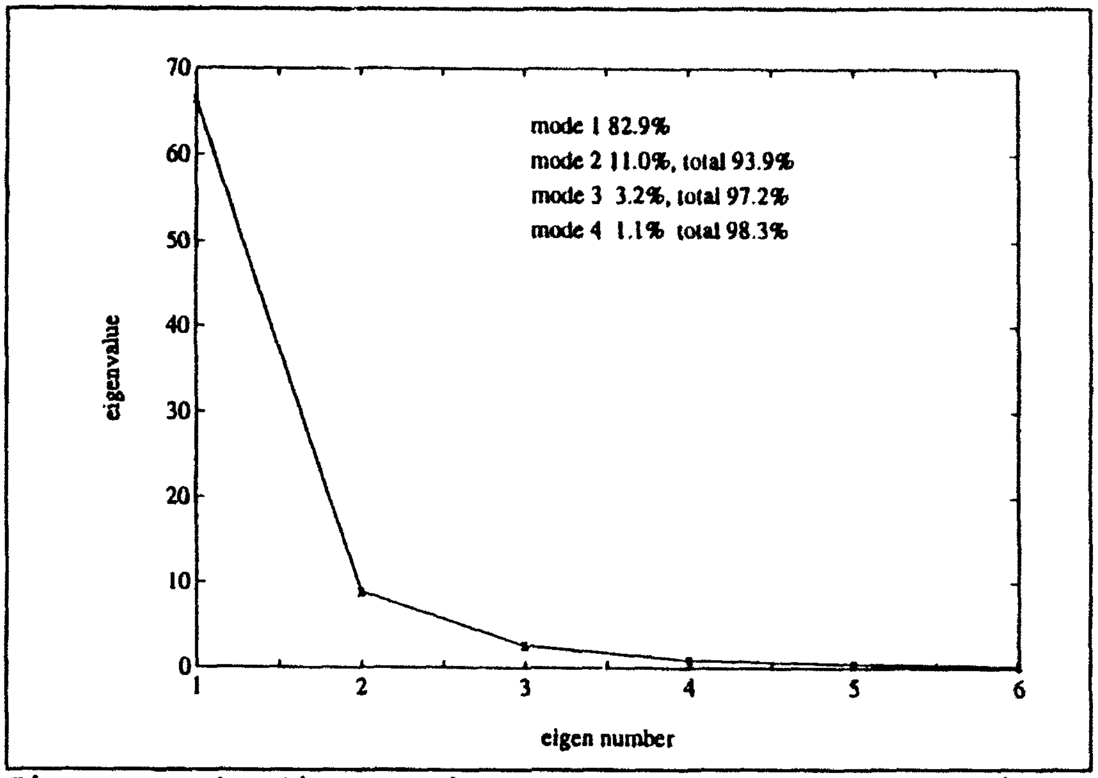

Figure 19 The first 6 eigenvalues. Percentage of variance is shown for first 4 modes.

B. OBJECTIVE ANALYSIS

Each XBT, after the application of the EOF decomposition, is represented by four modal amplitudes. The problem is to interpolate the modal amplitudes to arbitrary positions within the analysis region using objective analysis. It was decided to compute the interpolated modal amplitudes at regular intervals using a half degree spacing in both longitude and latitude, over a grid extending from 37 to 40 degrees North and 64 to 72 degrees West. The length scale between grid points of approximately 50 km was chosen because

it is comparable to the Rossby radius of deformation at this latitude.

The estimate, $\hat{X}$, of each amplitude at each grid location using equation 31 was computed. The correlations are assumed to depend solely upon the distance between observations and similarly between observations and grid points.

1. Determination of spacial correlation matrices

The distance between XBT sites was calculated and grouped into bins. Several bin intervals were considered with the object being to find an interval that gave a reasonable number of data pairs per bin, allowing an unbiased measure of correlation by distance to be determined. This was achieved using the program DEEPCOR and the subroutine CALC described in Appendix 3B.

The number of data pairs for the three intervals used are shown at Table I. The 25 km interval gave sufficient data pairs for each bin out to a distance of 200 km, and allowed eight spatial correlation estimates to be made. The results are shown in Figures 20-23.

The correlation function described in equation 33 was used to model the correlation estimates shown in Figures 20-23. The parameter a is equal to the distance at which the correlation falls to zero, and is given as the point where the curve in Figures 20-23 crosses the X axis. The value of b is the value of the distance when the correlation equals the e folding distance (e^{-1}). The value of the coefficients a and b

were found iteratively using the program FUNCTION given in Appendix 4B. In this program, the square error (ϵ),

$$\epsilon=(C_r-C_n)^2$$

equation 37

between the original data points, C_n, in Figures 20-23, and the iterated values C_r, calculated using equation 33, is minimized. The iteration sequence is intialized with values of a and b from visually inspecting Figures 20-23.

In order to determine whether each incremented value of a and b should be larger or smaller than the initial value, equation 33 was differentiated with respect to a and with respect to b. The analytical solution was used to increment a and b in such a way that the mean square error was reduced with each iteration. The iteration was repeated until the error had reduced to 0.05. The final values of the parameters a and b are shown in Table II.

It is assumed that the distance correlation function determined above for C_θ will also be applicable in the observation to grid point correlation matrix $C_{x\theta}$.

The objective analysis FORTRAN source programs are provided in Appendix 5B, 6B and 7B for reference. The first guess for each analysis is taken as the local weighted average of the modal amplitudes. Output from the objective analysis

consists of contour maps of the first four modal amplitudes, and analysis error of the interpolated amplitudes.

Contour plots of the first four modal amplitudes are shown in Figures 24-27 and their associated error maps in Figures 28-31.

2. The reconstruction

Using the modal amplitudes calculated by the objective analysis, synthetic XBTs were reconstructed at each of the grid points using Equation 16. However, the error in the XBT reconstruction is dependent on the position of the reconstruction. Synthetic XBTs produced in areas with high concentrations of observation stations are expected to suffer less error in reconstruction than synthetic XBTs produced in areas with sparsely populated data. The error variance in each XBT was calculated using equation 34 and the resulting error variance map is shown in Figure 32.

C. REDUCING THE NUMBER OF XBTs

Of ultimate interest is the size of the error variance in XBTs reconstructed within the analysis area. From the associated error variance map it is possible to assess, for any given position, the value of reconstructing and using a synthetic XBT at that point.

It was decided that for a reconstructed synthetic XBT, less than 30% error could be of use. The area inside the 30% contour of Figure 32 was noted. Successive XBTs were removed

and the objective analysis repeated until the 30% contour became the central or first contour. This meant that the area that was now enclosed represented error variances greater than 20% but less than 30%. The number of XBTs remaining was noted.

The original XBTs were numbered sequentially and the FORTRAN program RANDUM was used to place these numbers in random order. On commencing the objective analysis suite of programs, subroutine REDUCE permitted the number of XBTs to be used in the objective analysis to be varied.

Table I NUMBER OF DATA POINT PAIRS PER BIN FOR THREE DIFFERENT BIN SIZES.

12 km bin size									
12	24	36	48	60	72	84	96	108	120
81	104	114	142	184	190	214	226	210	168
25 km bin size									
25	50	75	100	125	150	175	200	225	250
195	266	412	452	370	310	276	224	90	68
50 km bin size									
50	100	150	200	250	300	350	400	450	500
461	864	680	500	158	58	52	62	92	106

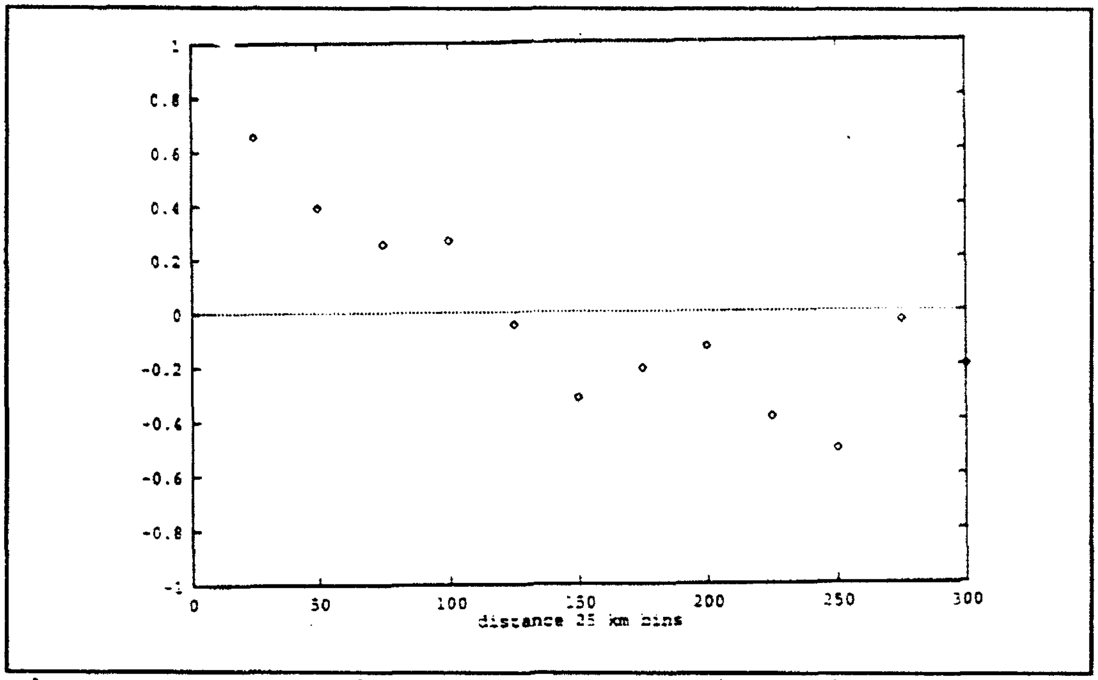

Figure 20 Correlation between data point pairs using a 25 km distance bin for the first modal amplitude.

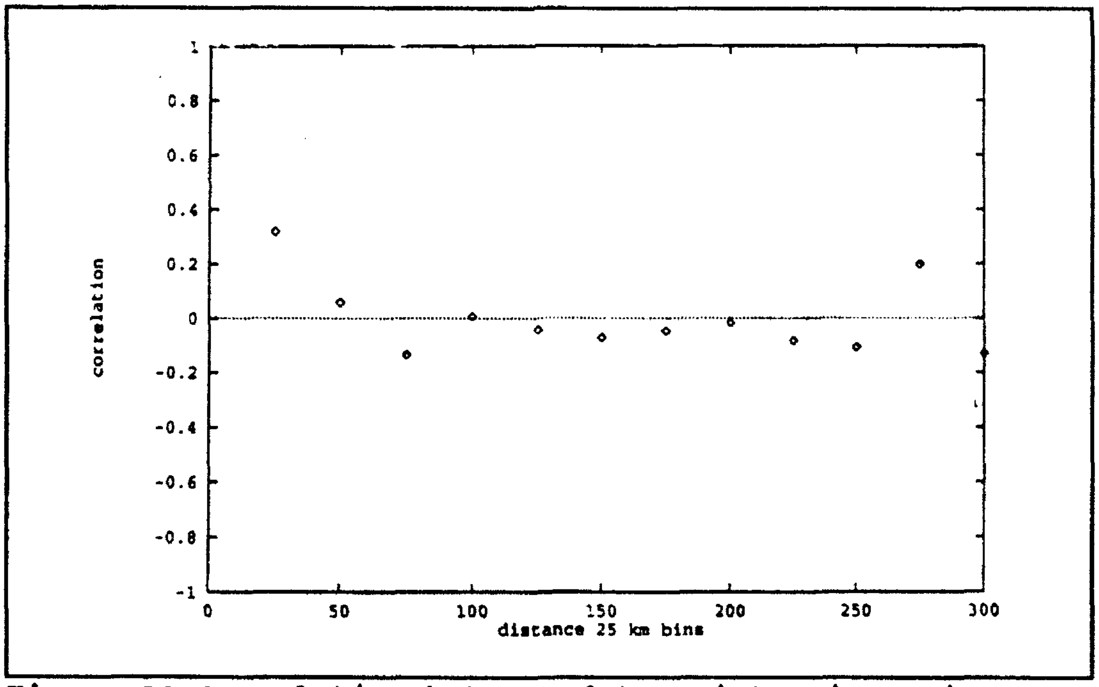

Figure 21 Correlation between data point pairs using a 25 km bin for the second modal amplitude.

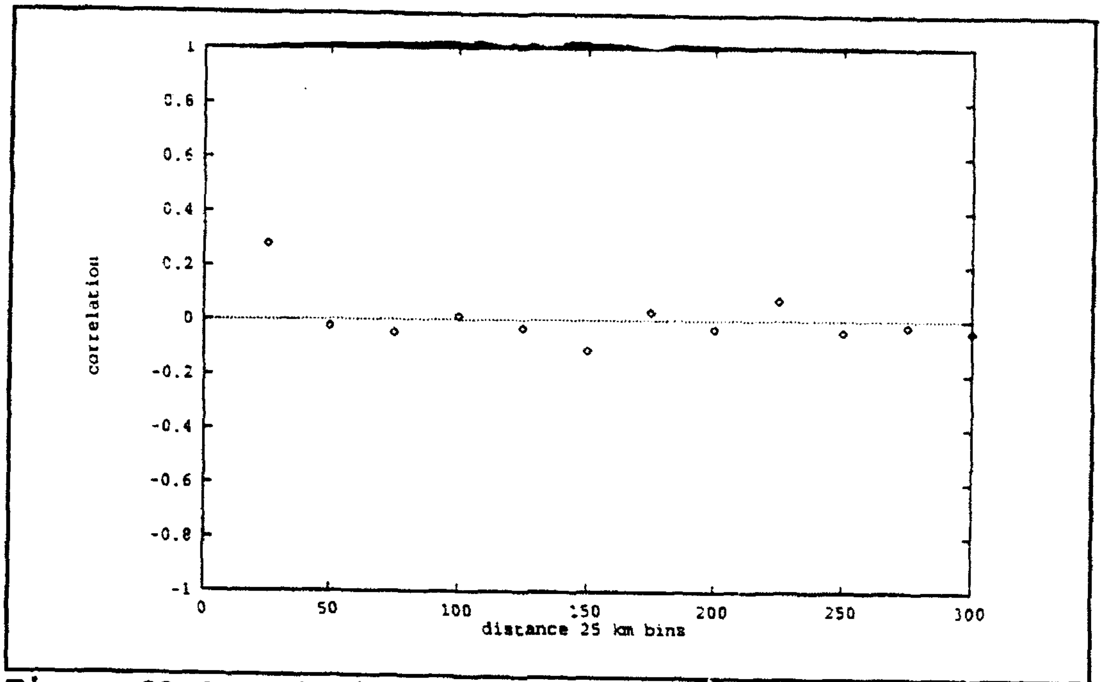

Figure 22 Correlation between data point pairs using a 25 km bin for the third modal amplitude.

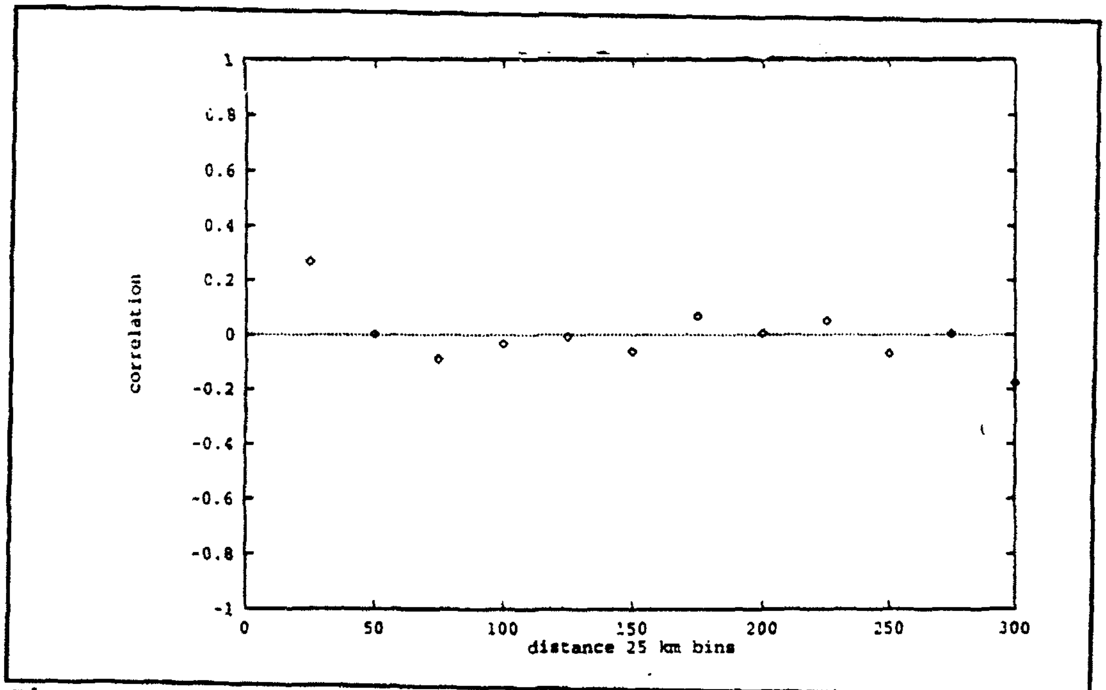

Figure 23 Correlation between data point pairs using a 25 km bin for the fourth modal amplitude.

Table II PARAMETER VALUE a AND b FOR EACH OF THE MODAL AMPITUDES.

	MODAL AMPLITUDES			
	1	2	3	4
a	134	65	37	44
b	64	25	28	26

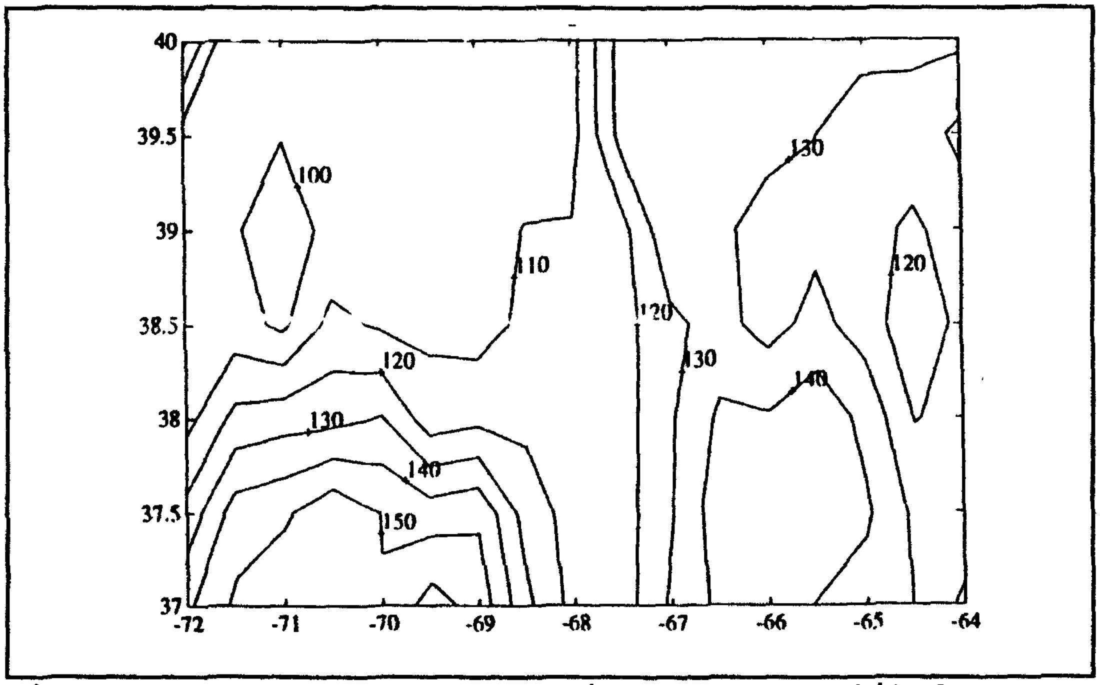

Figure 24 Contour map of the first modal amplitude.

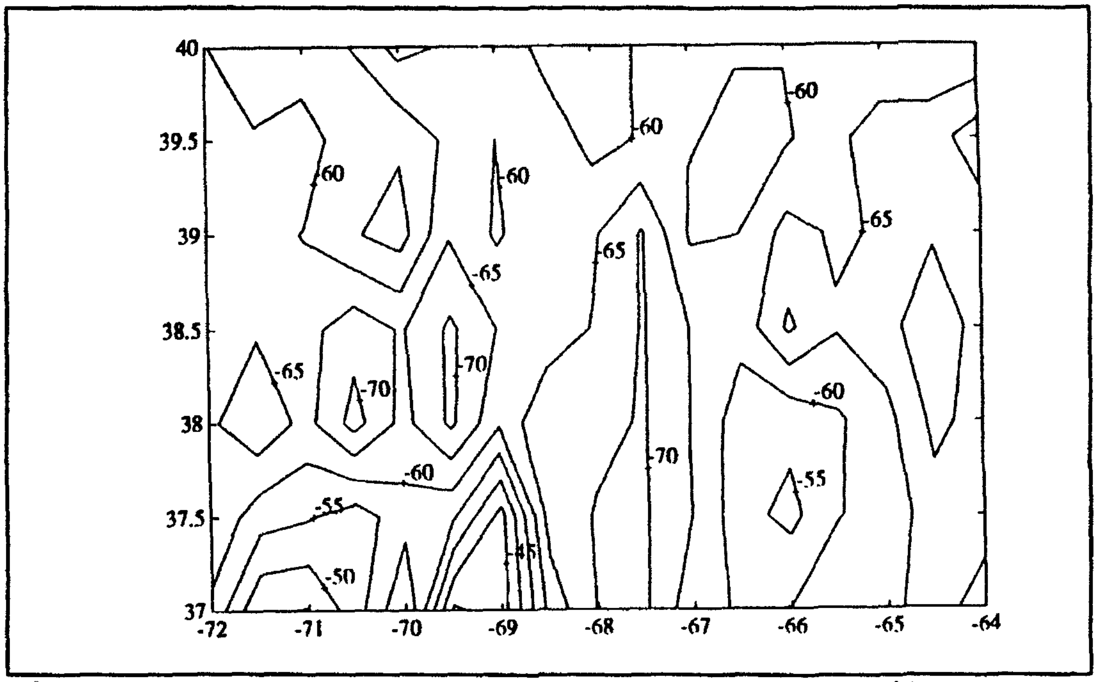

Figure 25 Contour map of the second modal amplitude.

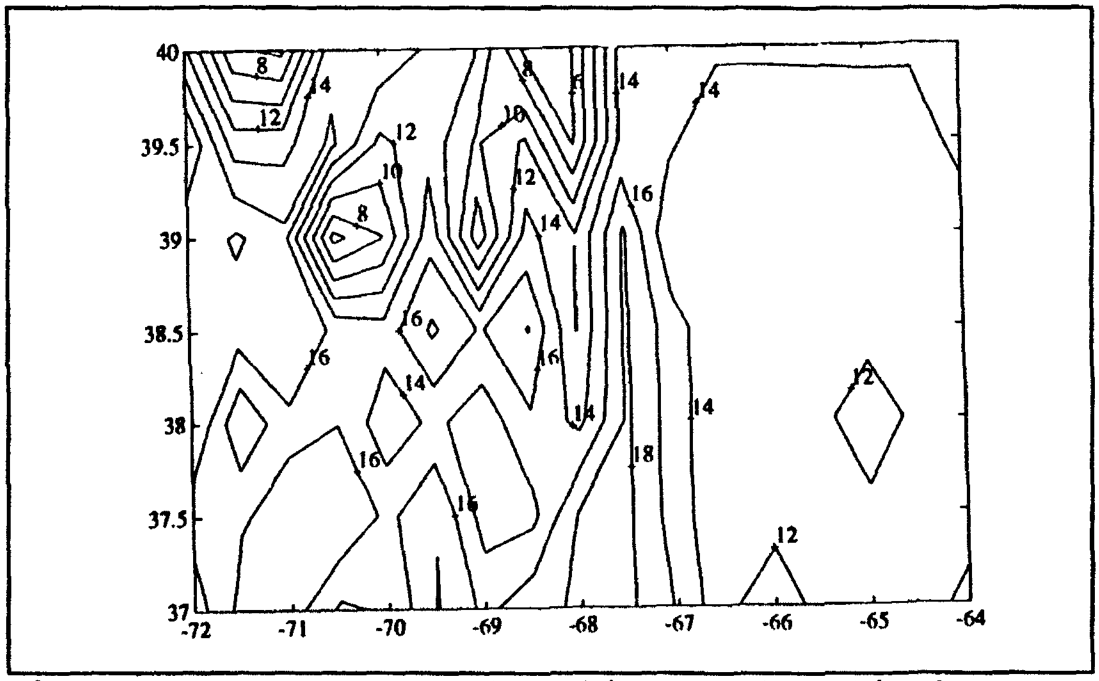

Figure 26 Contour map of the third modal amplitude.

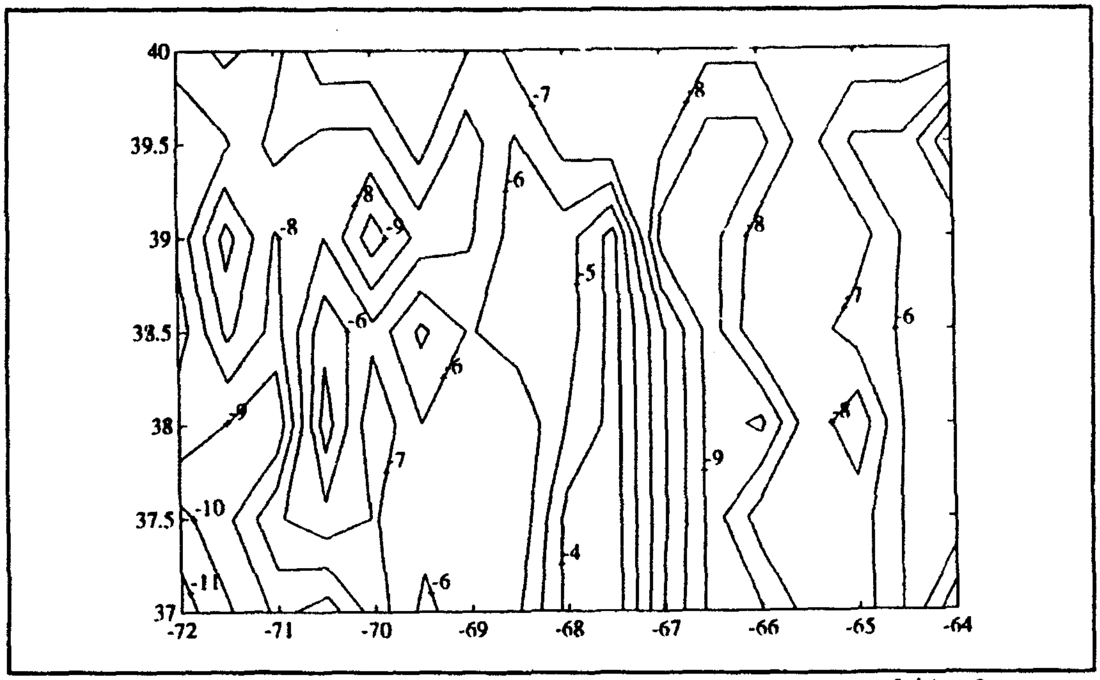

Figure 27 Contour map of the fourth modal amplitude.

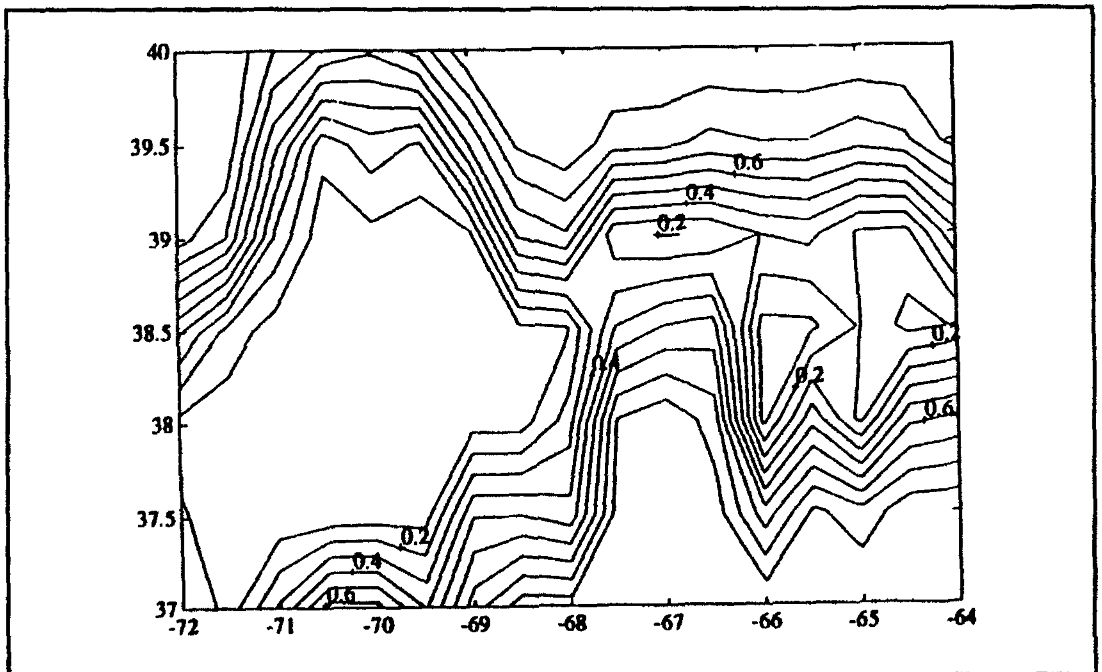

Figure 28 Error variance map as a result of interpolating the first modal amplitudes. 0.1 contour intervals.

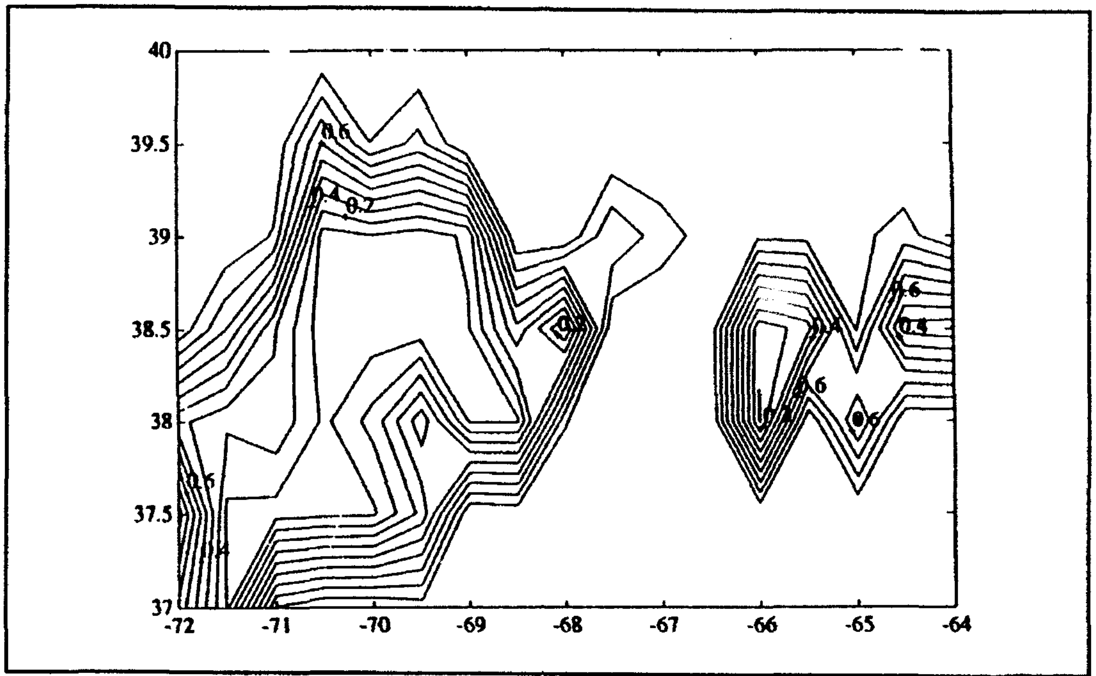

Figure 29 Error variance map as a result of interpolating second modal amplitudes. 0.1 contour intervals.

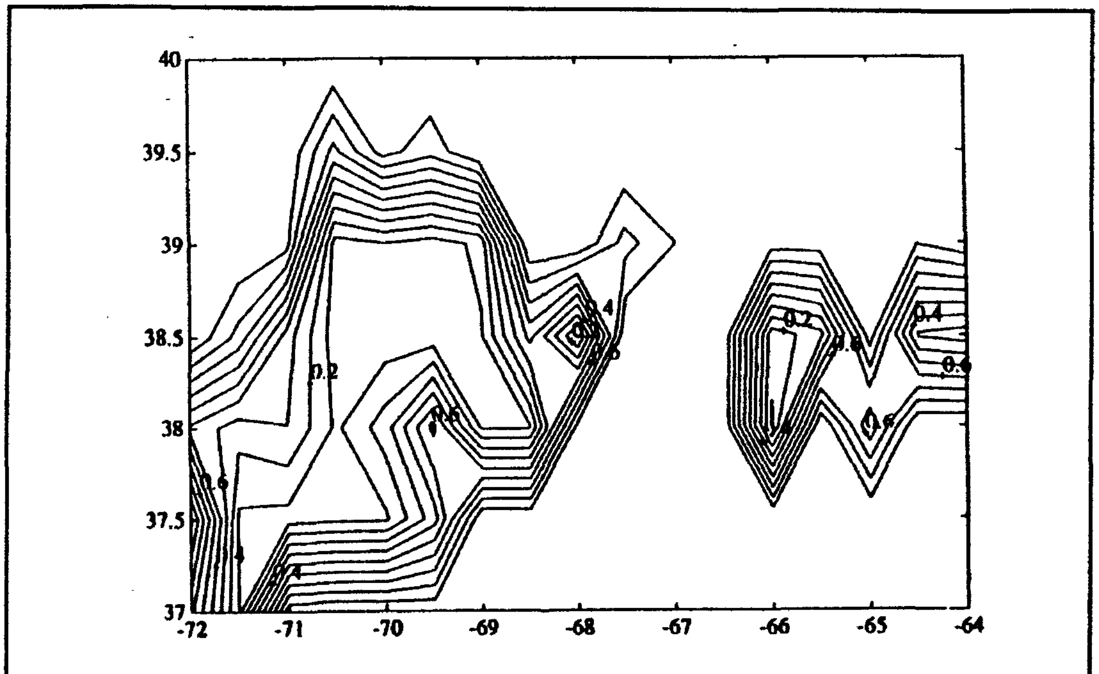

Figure 30 Error variance map as a result of interpolating third modal amplitudes. 0.1 contour intervals

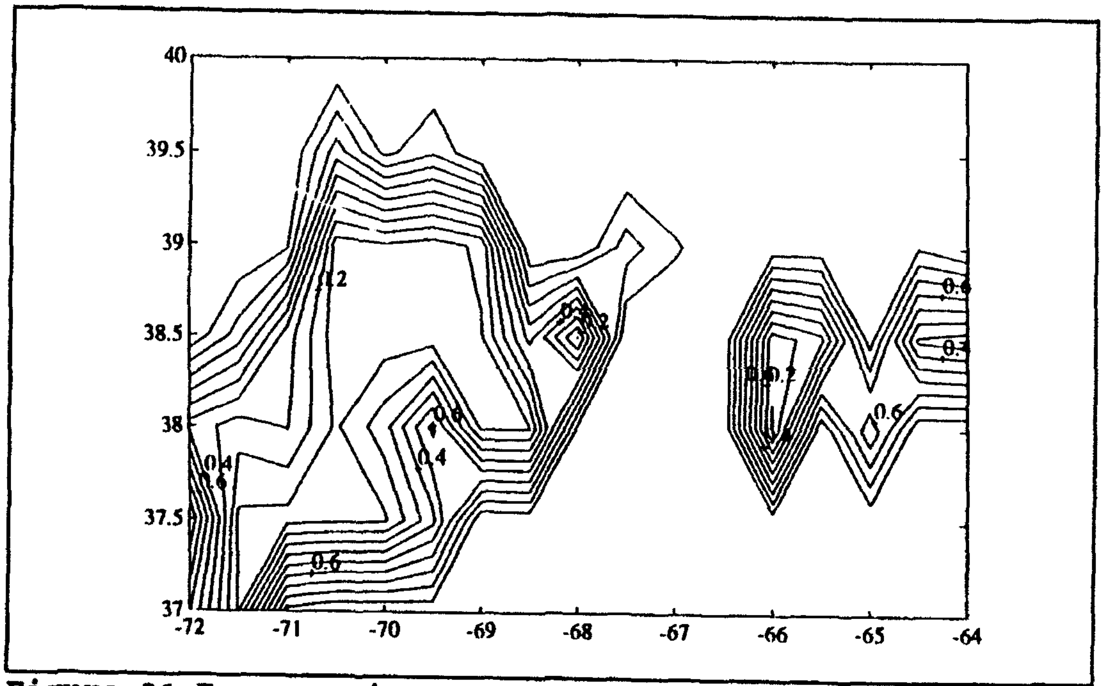

Figure 31 Error variance map as a result of interpolating the fourth modal amplitudes. 0.1 interval contour spacing.

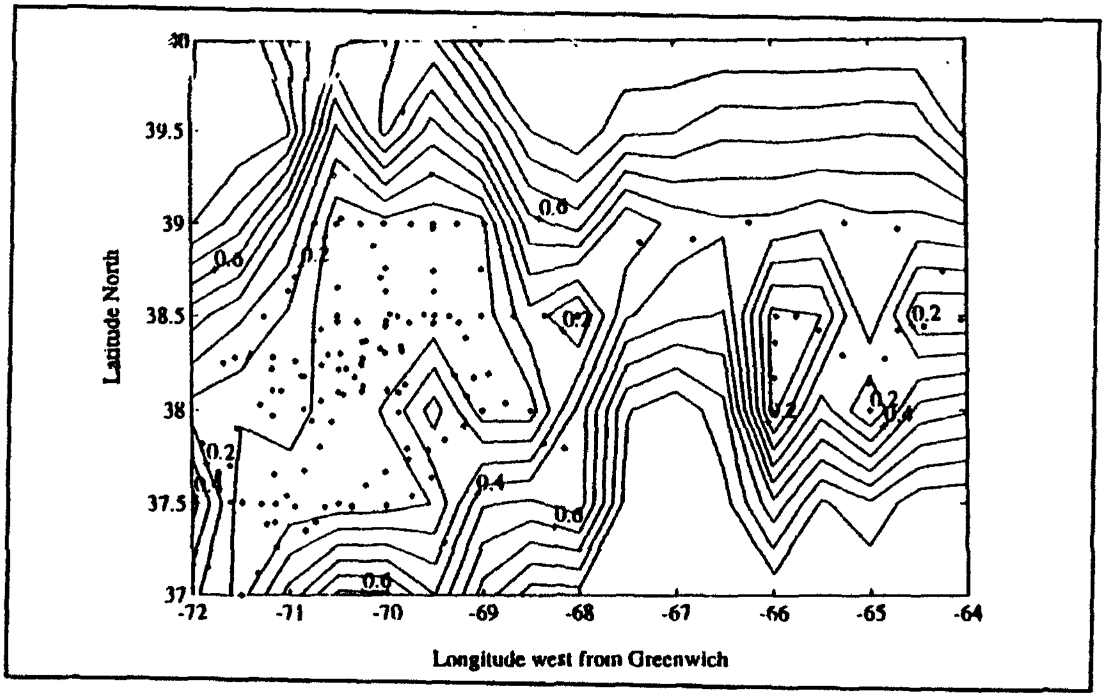

Figure 32 Error variance map, produced by using equation 37, effectively the combination of Figures 28-31. The * indicates the original XBT sites. The contours represent the amount of confidence that can be placed in a reconstruction at any location within the area.

V INITIAL ANALYSIS

A. RECONSTRUCTION ALONG A LINE OF LATITUDE

To get a feel for how good or bad the reconstructions appeared, a line of synthetic XBTs along 37.5 N from 72 West to 69.0 West were reconstructed at 1/2 degree intervals and are shown in Figures 33-36. A group of real XBTs, taken along the proximity of this line are shown in Figures 37- 44. The positions of the real XBTs are readily apparent by consulting Figure 45 which indicates the position of the XBTs used in this analysis.

Although the two groups of diagrams show a general similarity in shape there are enough differences to cause concern. Firstly, the synthetic XBTs all tend to exhibit an temperature minimum at about 200 metres that is more exaggerated than in the real XBTs surrounding this line of latitude. Secondly, the synthetic XBTs also show a strong negative temperature gradient within the first 30 to 80 metres that again is not apparent in most of the real XBTs, which for the most part are isothermal or exhibit only a slightly negative temperature gradient over the same depth range.

B. RECONSTRUCTION OF ONE XBT

To pursue these discrepancies further XBT 7105A was selected for closer study. This particular XBT was chosen because its position is at the same location as a synthetically produced XBT. Thus, it would be expected that the profiles of the real and the synthetic XBTs should show a very high degree of similarity. The real XBT 7105A, is shown in Figure 42 and the synthetic XBT in Figure 34. Again, the synthetic profile exhibits a temperature minimum at two hundred metres and a negative gradient in the surface layer, both features being less pronounced in the observed profiles.

As a check to ensure that the EOF decomposition had been performed correctly it was decided to reconstruct 7105A using all 80 modes. The results of this are shown in Figure 46 where comparison with the original and the synthetic XBT using four modes can be made directly. Figure 47 shows the difference between the original and the reconstructed XBT using 80 modes to be negligible, of the order of 10^{-6} degrees Celsius, whereas Figure 48, which shows the difference between the original and the reconstructed XBT using 4 modes, shows a much larger overall error of 0.44 degrees Celsius. Table III gives the mean square error for a selected number of modes.

C. RECONSTRUCTION AT CAST SITES

In addition, the objective analysis was performed at cast sites taken within the 10% error variance contour line of Figure 32. Thus, instead of the objective analysis being done on a regular grid, the procedure reconstructed synthetic XBTs at the same sites where the original XBTs had been taken. This was done as a check to ensure that the reconstructed error map was consistent and to gain a measure of how much error there was between the original XBTs and the synthetic reconstruction. A selection of these XBTs are shown in Figures 50 - 54, along with a graph of their associated RMS error. The position of the original casts can be found from Figure 49.

The RMS error at each of the 80 depth setting is computed as a percentage of the temperature value compared to the reconstruction using 4 modes. An average error, expressed as a percentage, is then obtained for each XBT, and the results are averaged over the set of XBTs used in the analysis. The overall error between the synthetic XBTs compared to the reconstruction using the four original EOFs was 5%, well within the 10% boundary.

Reconstruction of all 156 XBTs, using only four modes, gave an error, when compared to the original XBTs, of between 6-7%. The overall error between the OA reconstructions and the original XBTs was found to be between 10 and 11%.

D. RECONSTRUCTION AT SELECTED GRID POINTS

A selection of XBTs were reconstructed at grid point sites and the error compared to the originals that were likewise taken at the same points. These plots and the associated error graphs are shown in Figures 55 - 62. The position of each XBT is shown in Figure 63. XBTs 71 and 88, shown in Figures 60 and 61, are displaced from the nearest grid point (38.5 N 70 W), to which they are compared. These profiles are included to show the wide range of variability that exists within short spacial distances.

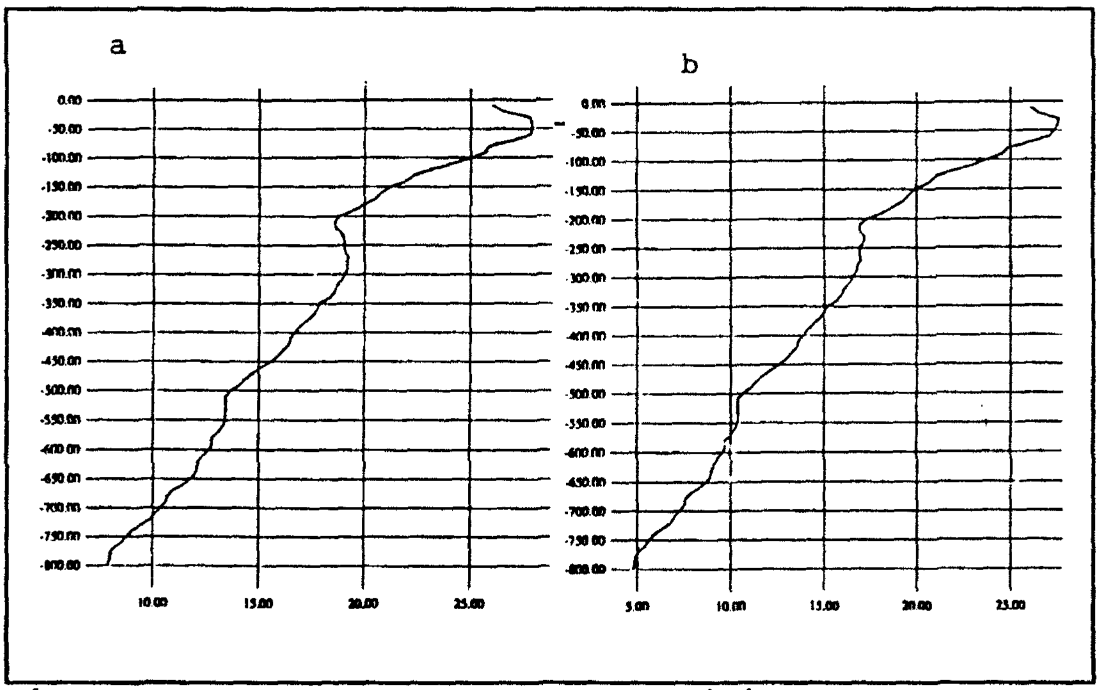

Figure 33 Reconstructed XBTs at positions 37.5 N 72 W (a)and 37.5 N 71.5 W (b).

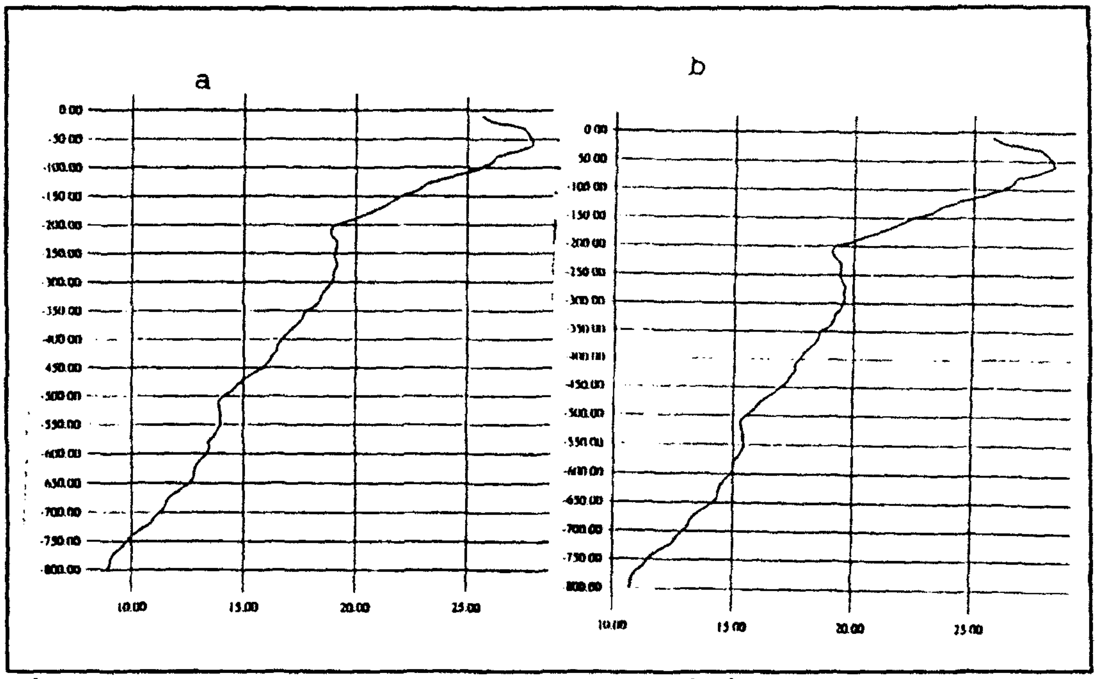

Figure 34 Reconstructed XBTs at positions 37.5 N 71W (a) and 37.5 N 70.5 W (b).

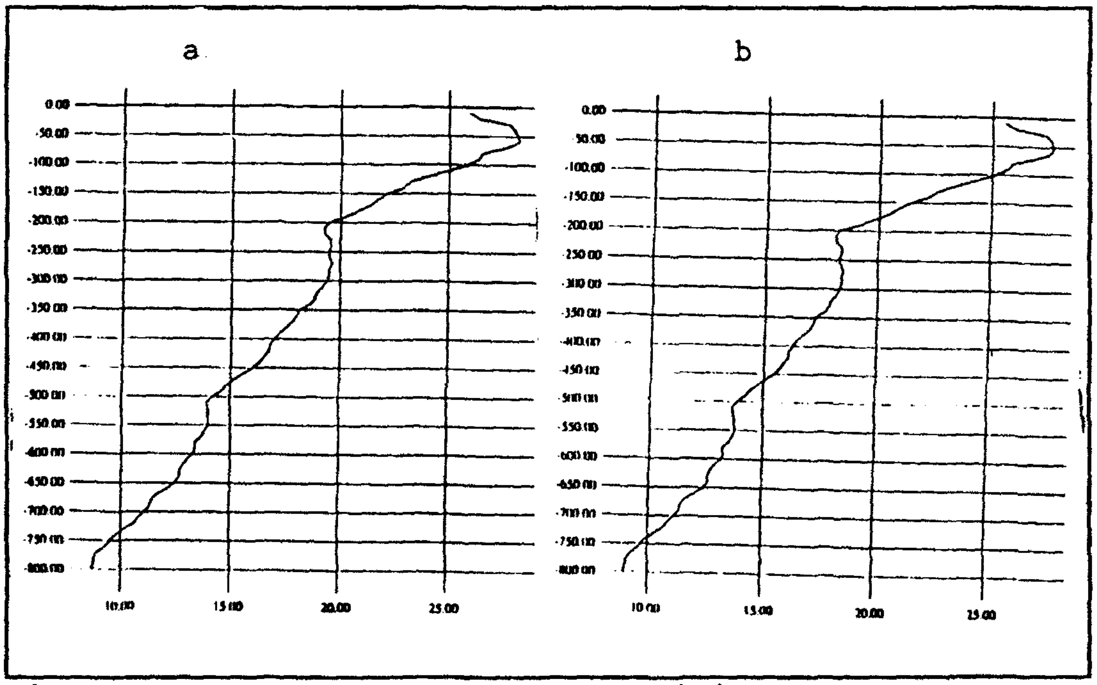

Figure 35 Reconstructed XBTs at positions 37.5 N 70 W (a) and 37.5 N 69.5 W (b).

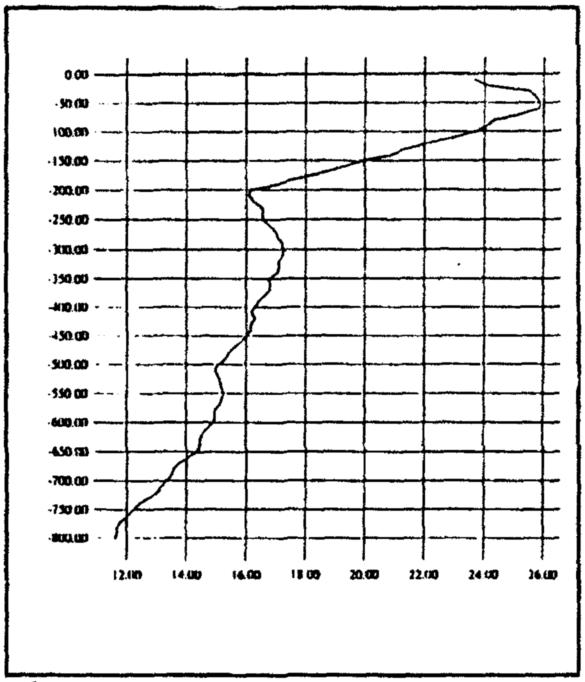

Figure 36 Reconstructed XBT at position 37.5 N 69 W.

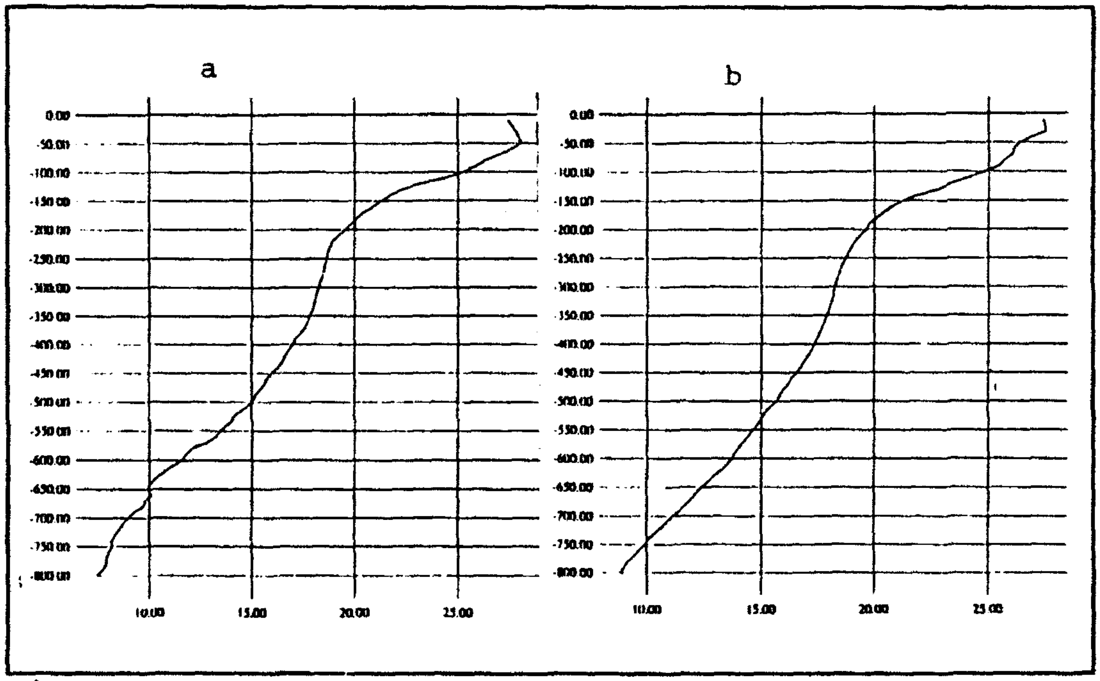

Figure 37 XBTs 766A (a) and 734A (b).

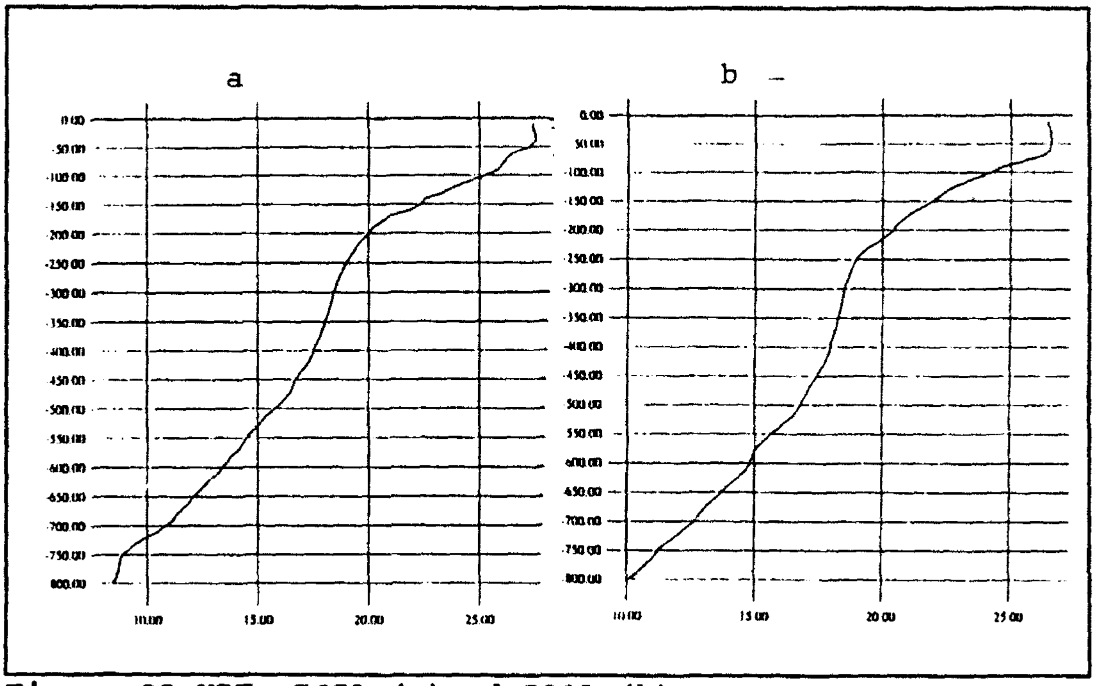

Figure 38 XBTs 765A (a)and 733A (b).

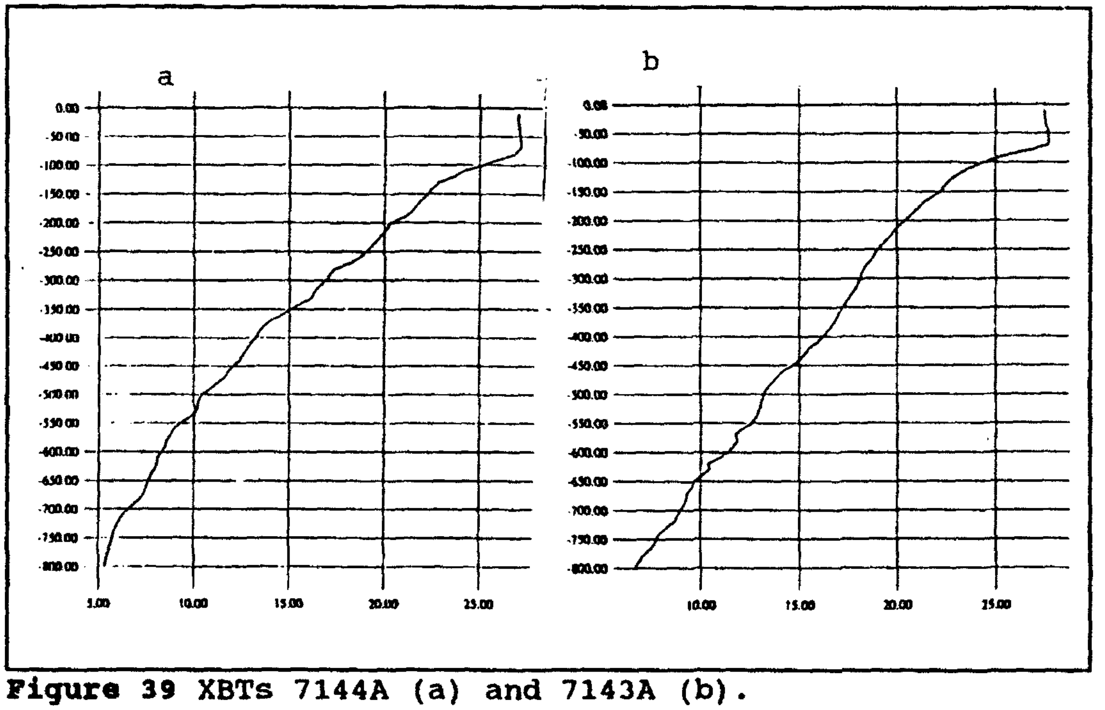

Figure 39 XBTs 7144A (a) and 7143A (b).

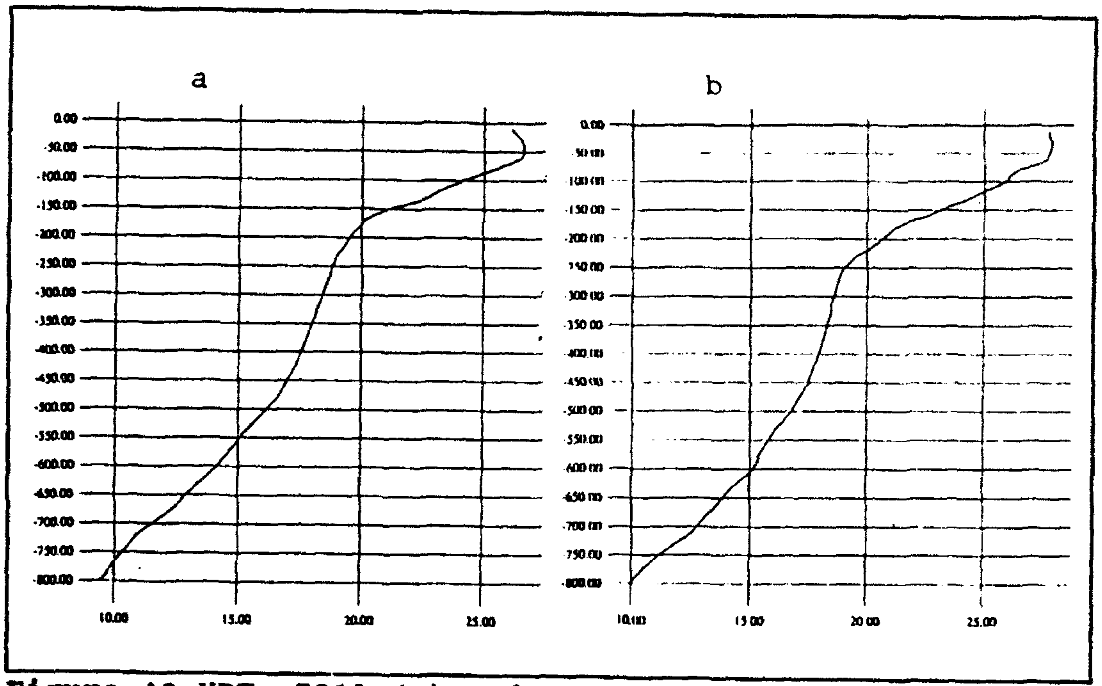

Figure 40 XBTs 731A (a) and 764A (b).

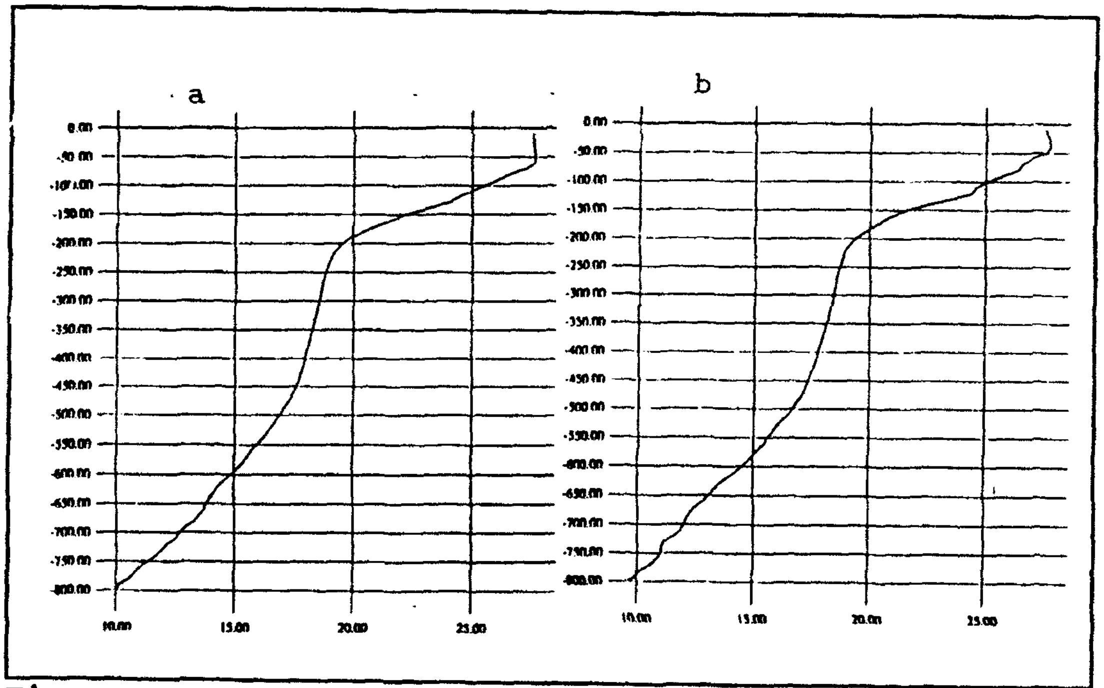

Figure 41 XBTs 7107A (a) and 7106A (b).

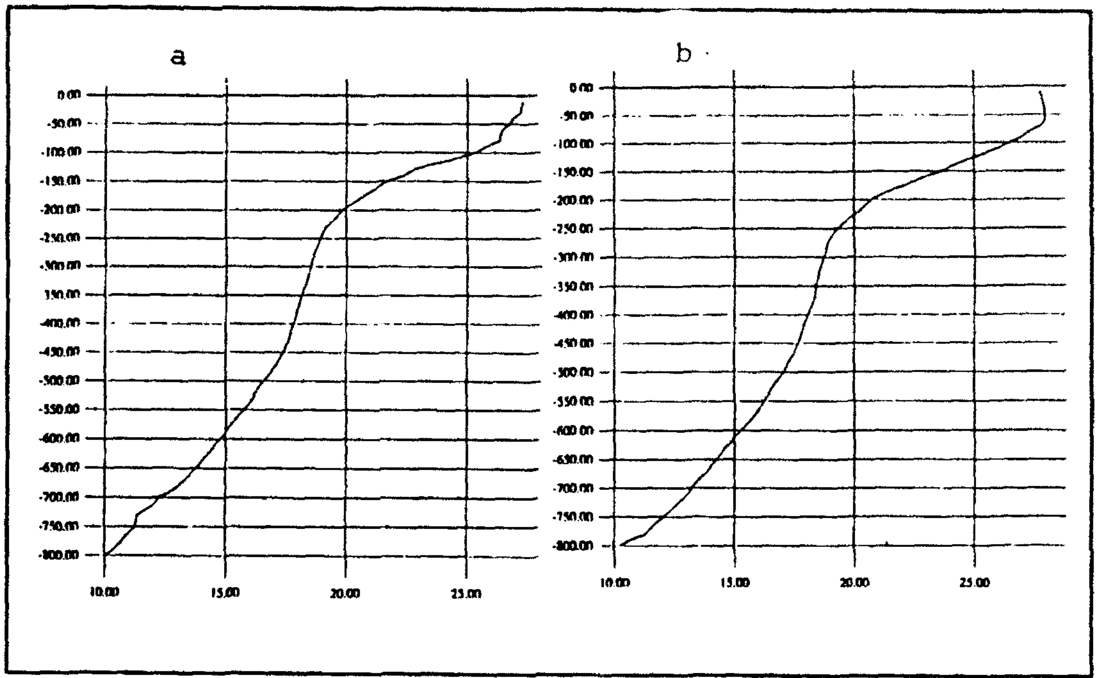

Figure 42 XBTs 7105A (a) and 763A (b).

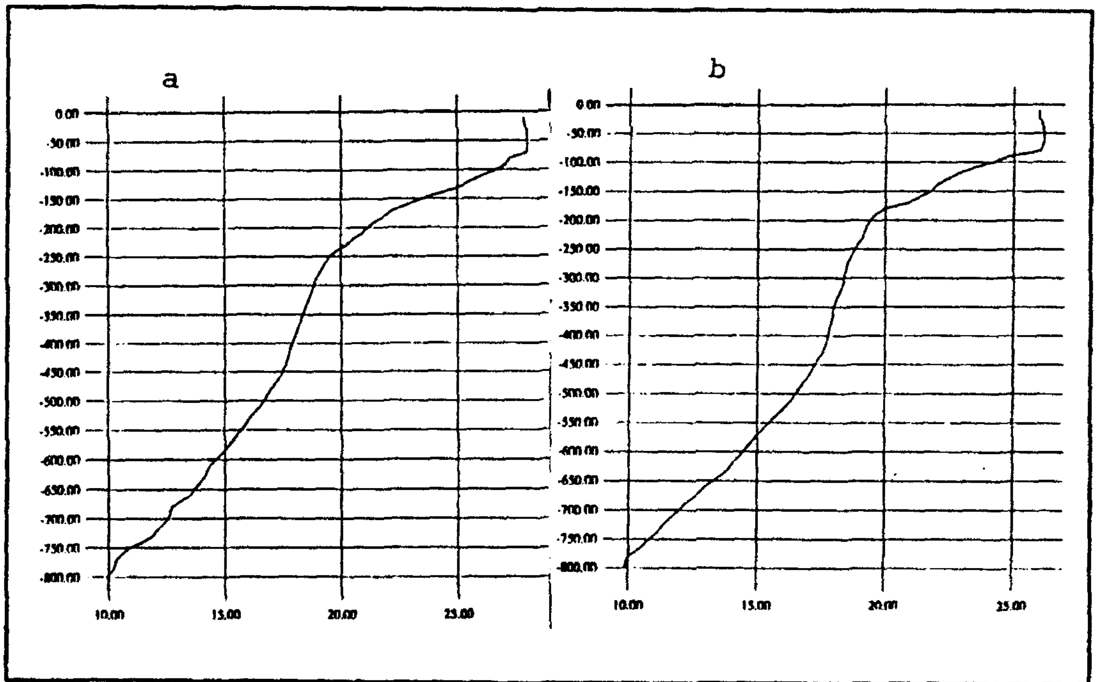

Figure 43 XBTs 762A (a) and 7104A (b).

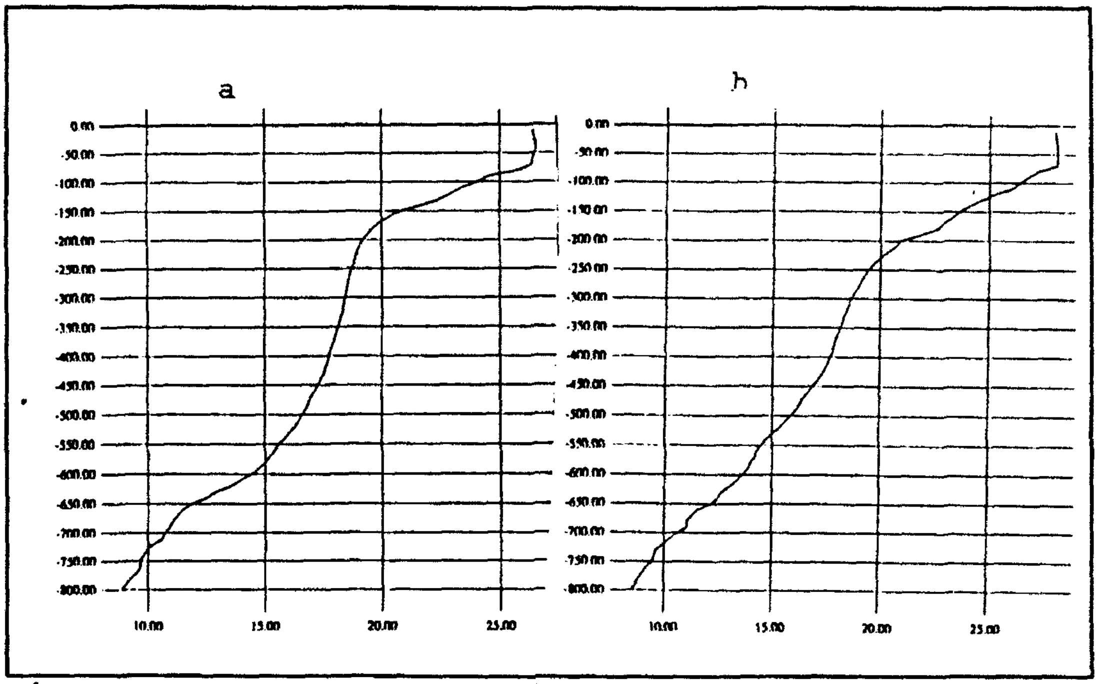

Figure 44 XBTs 7103A (a) and 761A (b).

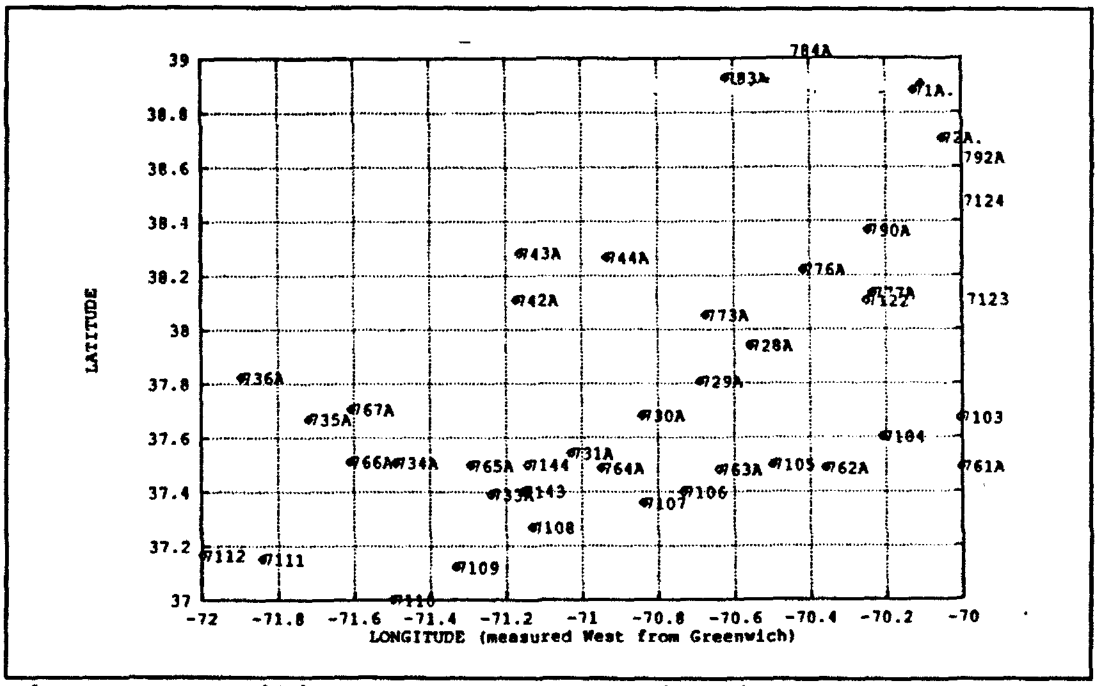

Figure 45 Positions of XBTs shown in Figures 33-44.

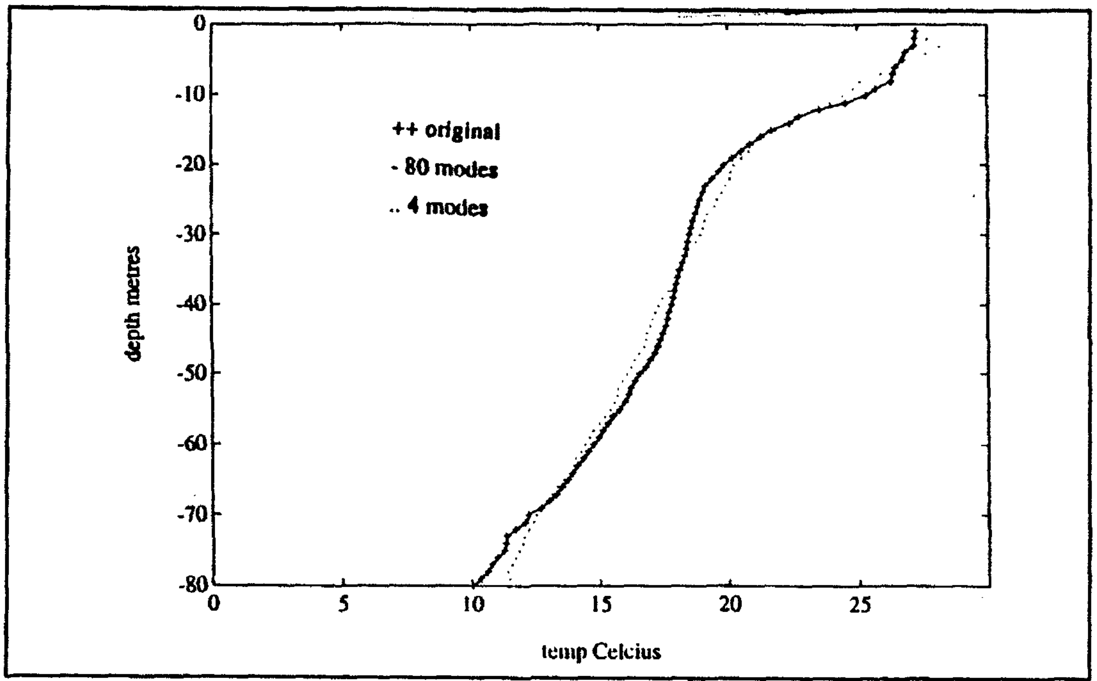

Figure 46 XBT 7105A showing original and reconstructions using 4 modes and 80 modes.

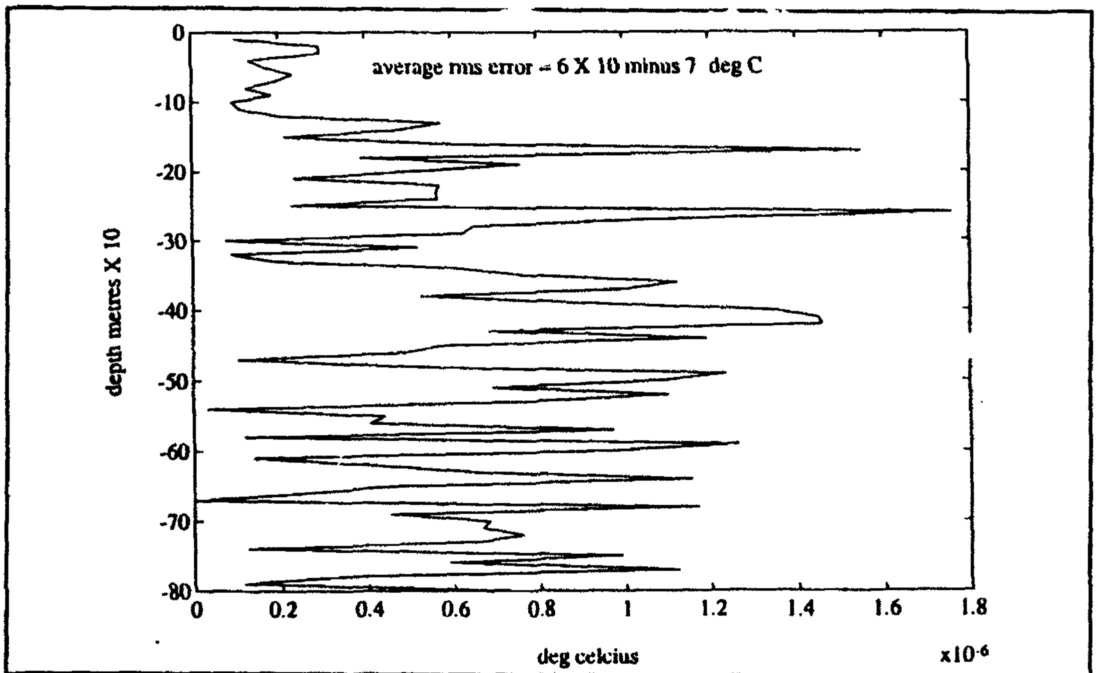

Figure 47 Graph showing difference between original temperature values and those produced through reconstruction of the XBT using all 80 EOF's.

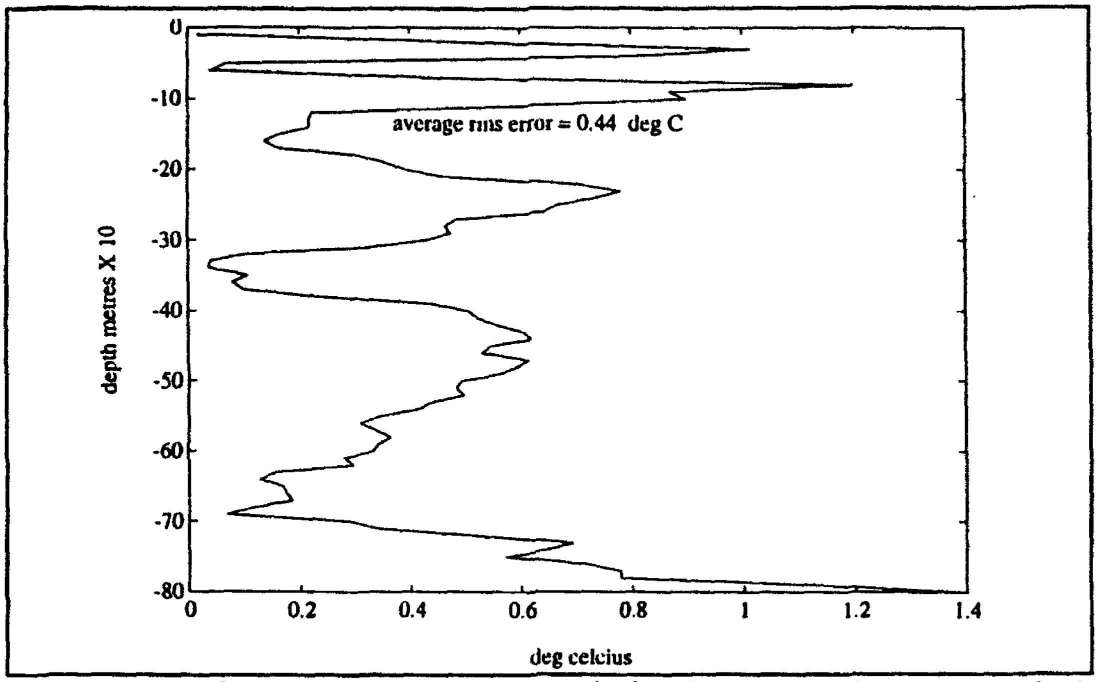

Figure 48 Difference between original temperature and that produced by reconstruction of XBT using only 4 modes.

Table III TABLE SHOWING MEAN RMS ERROR FOR ORIGINAL XBT 7105A COMPARED WITH ITS RECONSTRUCTION USING DIFFERENT NUMBERS OF MODES.

mode	1	2	3	4	5	6
error	4.4	1.6	0.8	0.4	0.3	0.3
mode	7	8	9	10	20	30.
error	0.2	0.1	0.1	0.12	0.07	0.04
mode	40	50	60	70	80	
error	0.03	0.02	0.01	0.001	0.001	

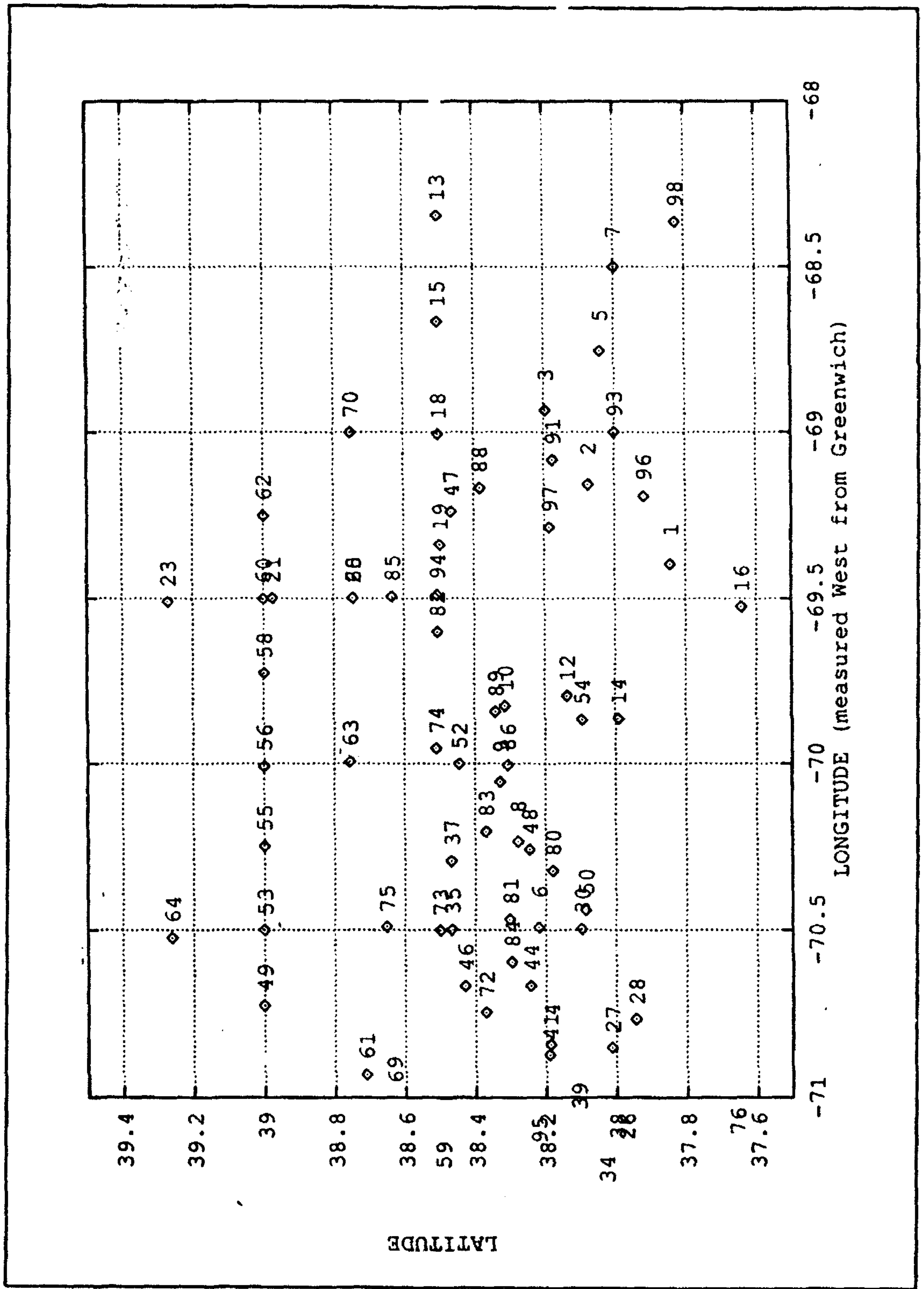

Figure 49 Sites used for selected reconstructions.

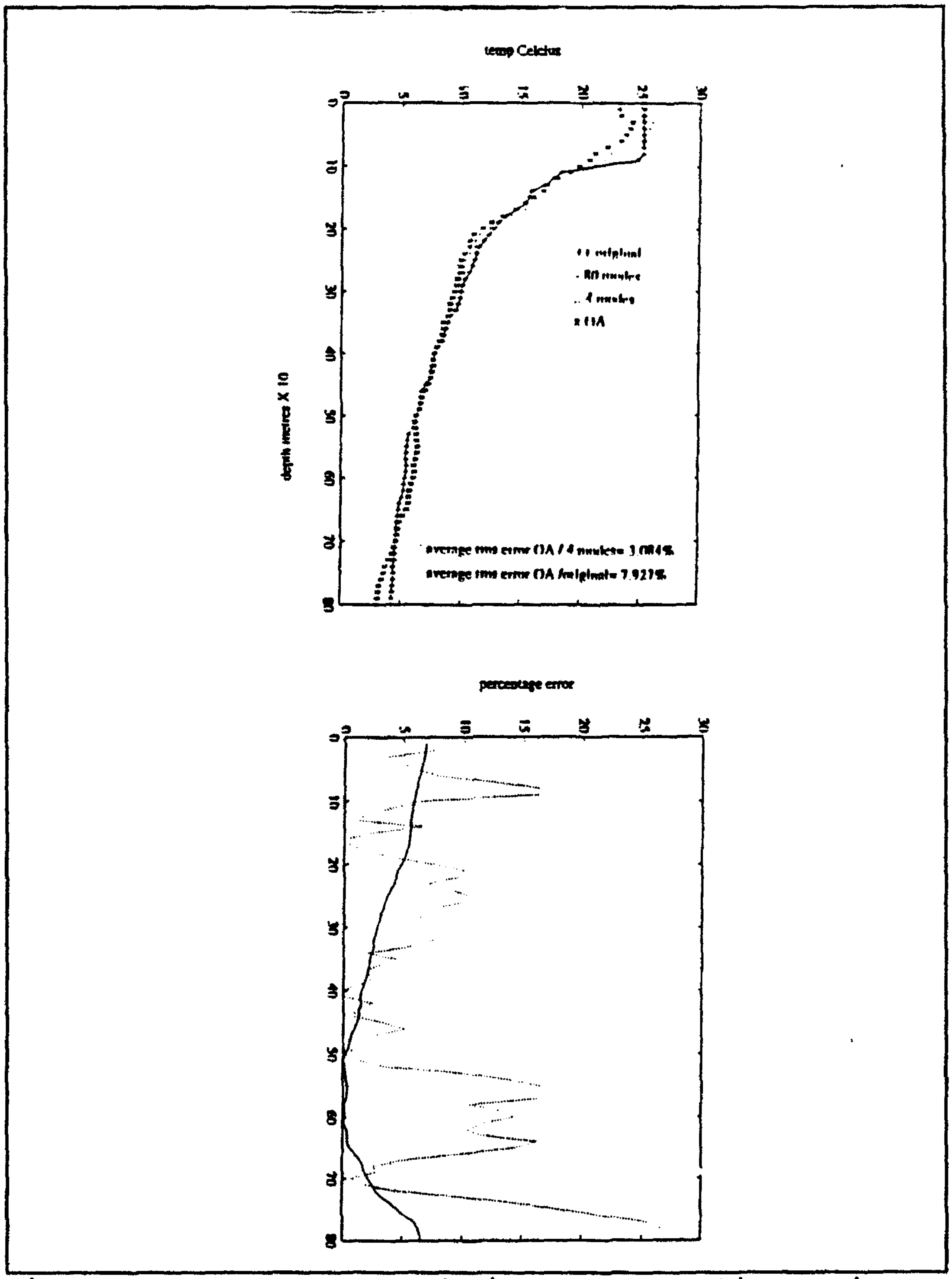

Figure 50 a, XBT 3 compared with reconstructions using 4 and 80 EOFs, the OA being performed onto the site of the XBT cast. b, rms error between the OA and the 4 and 80 mode reconstruction for all depths.

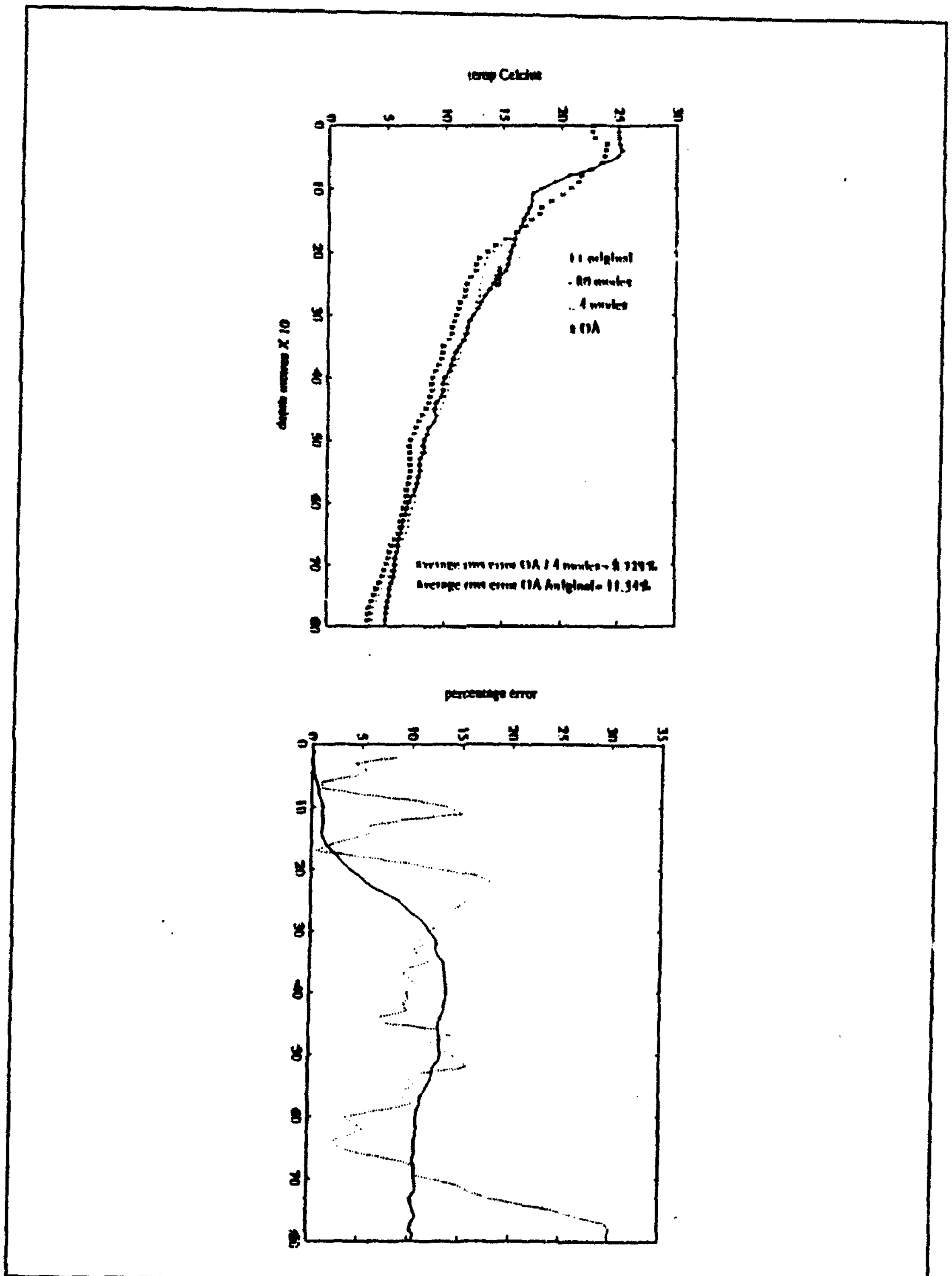

Figure 51 a, XBT 6 compared with reconstructions using 4 and 80 EOFs, the OA being performed onto the site of the XBT cast. b, rms error between the OA and the 4 and 80 mode reconstruction for all depths.

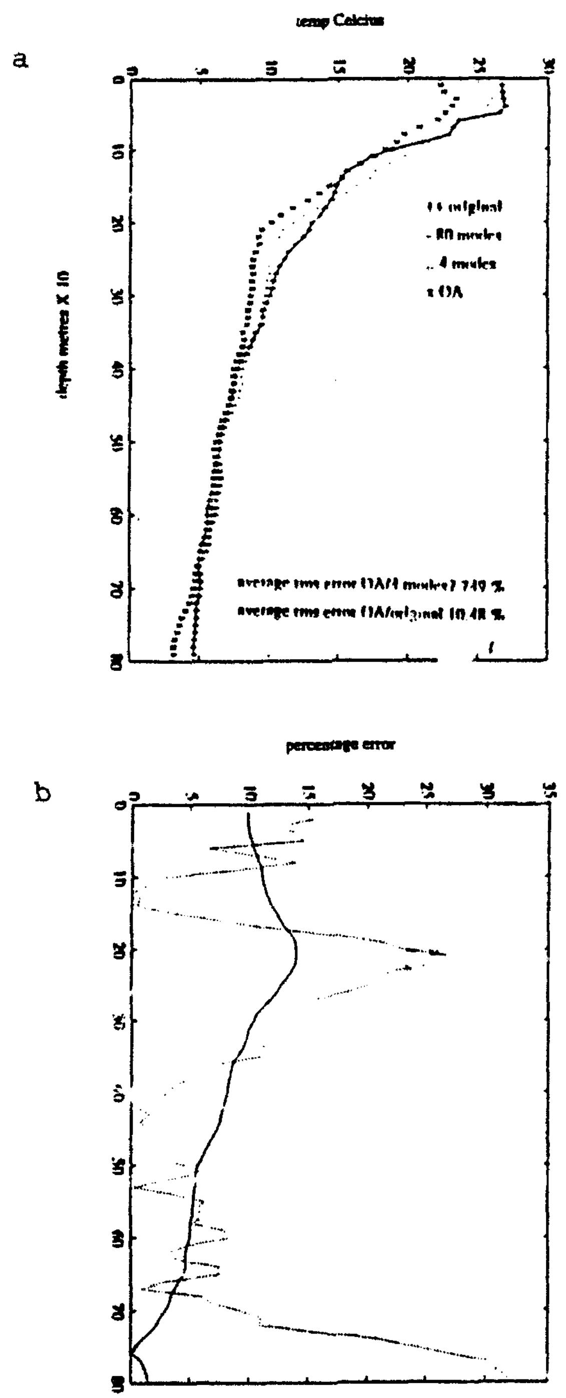

Figure 52 a, XBT 35 compared with reconstructions using 4 and 80 EOFs, the OA being performed onto the site of the XBT cast. b, rms error between the OA and the 4 and 80 mode reconstruction for all depths.

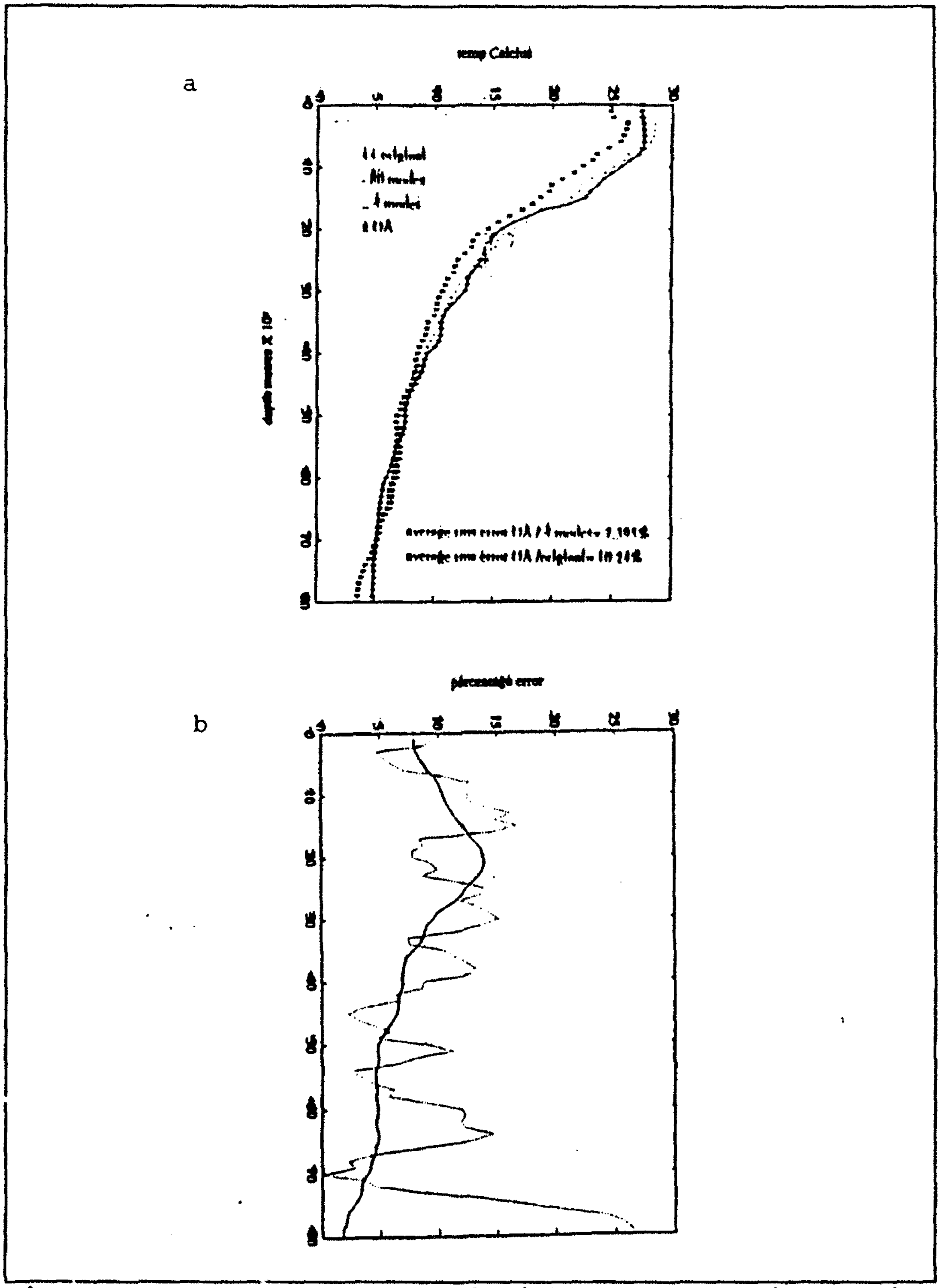

Figure 53 a, XBT 37 compared with reconstructions using 4 and 80 EOFs, the OA being performed onto the site of the XBT cast. b, rms error between the OA and the 4 and 80 mode reconstruction for all depths.

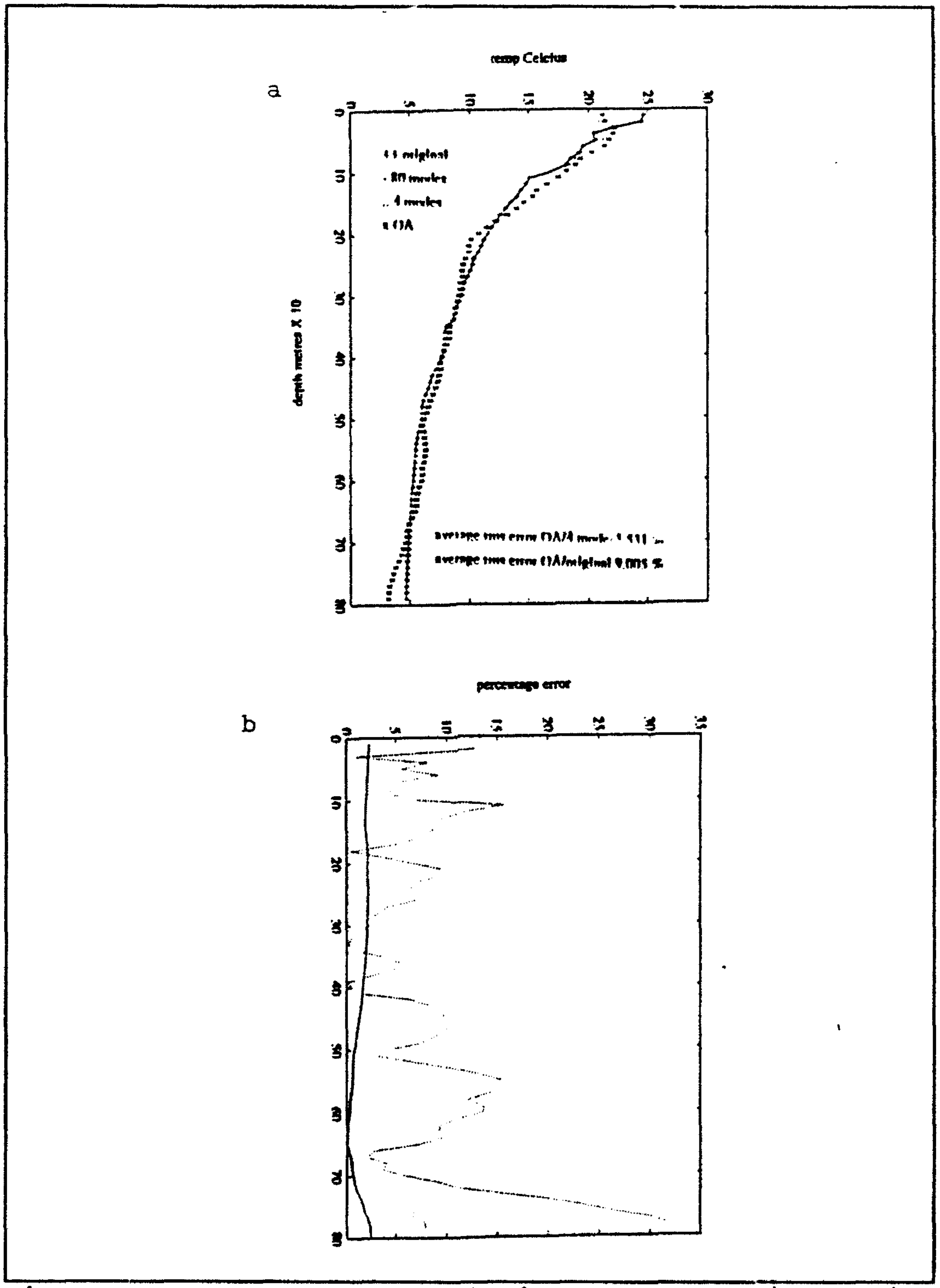

Figure 54 a, XBT 63 compared with reconstructions using 4 and 80 EOFs, the OA being performed onto the site of the XBT cast. b, rms error between the OA and the 4 and 80 mode reconstruction for all depths.

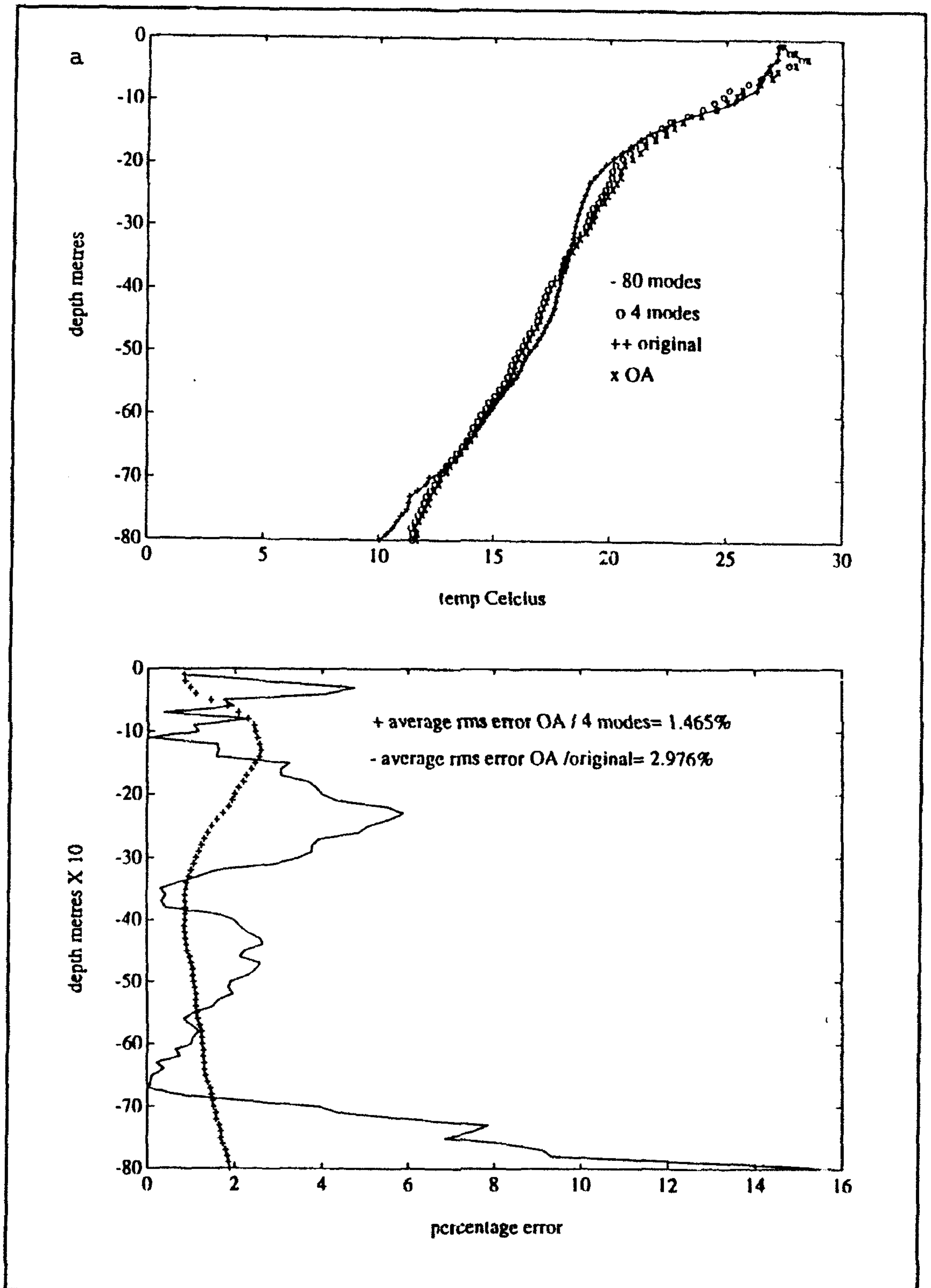

Figure 55 a, XBT 14 compared to 4 and 80 modes and to OA reconstruction. b, RMS error between OA, 4 modes and the original for each depth.

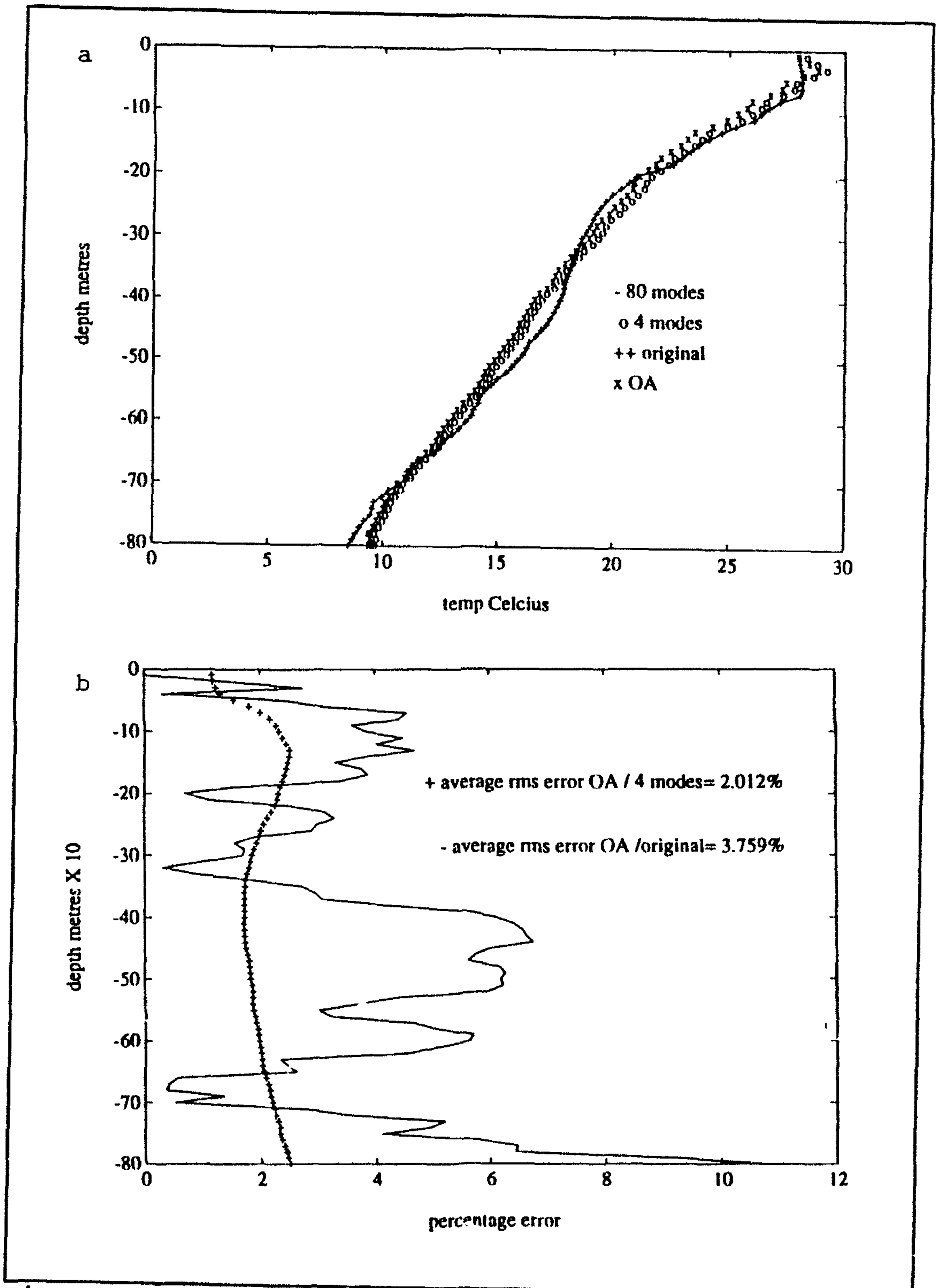

Figure 56 a, XBT 37 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

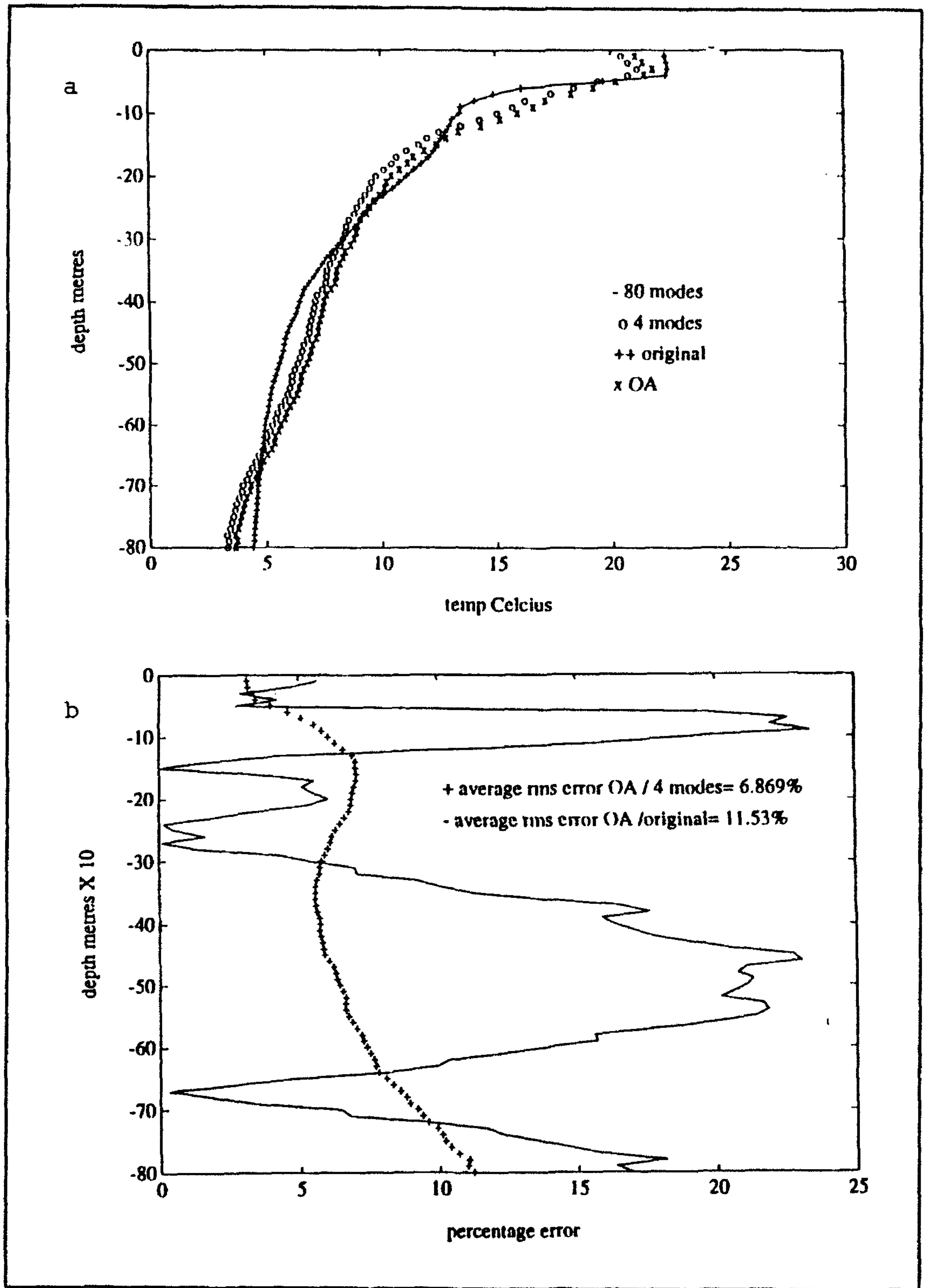

Figure 57 a, XBT 150 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

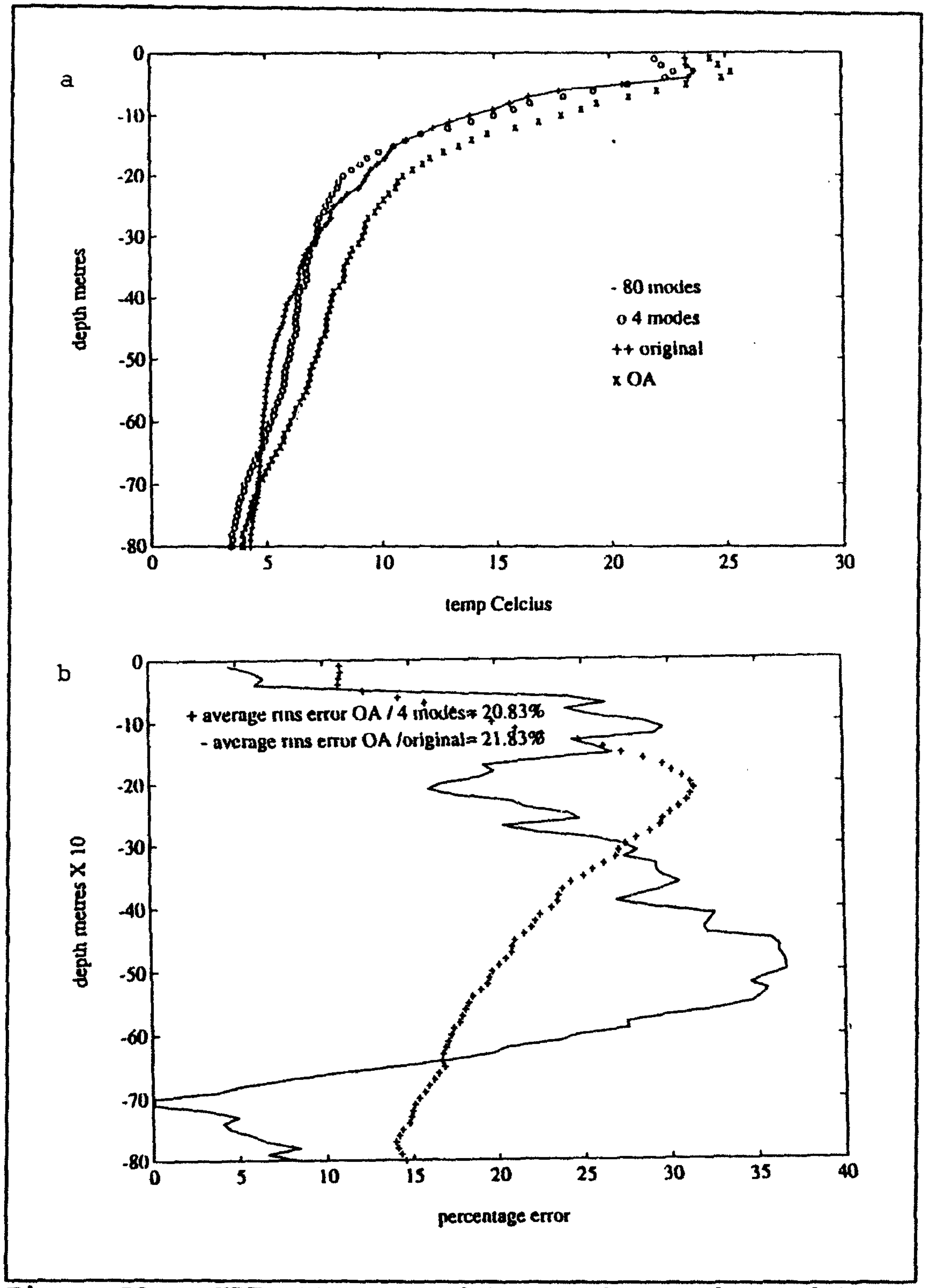

Figure 58 a, XBT 121 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

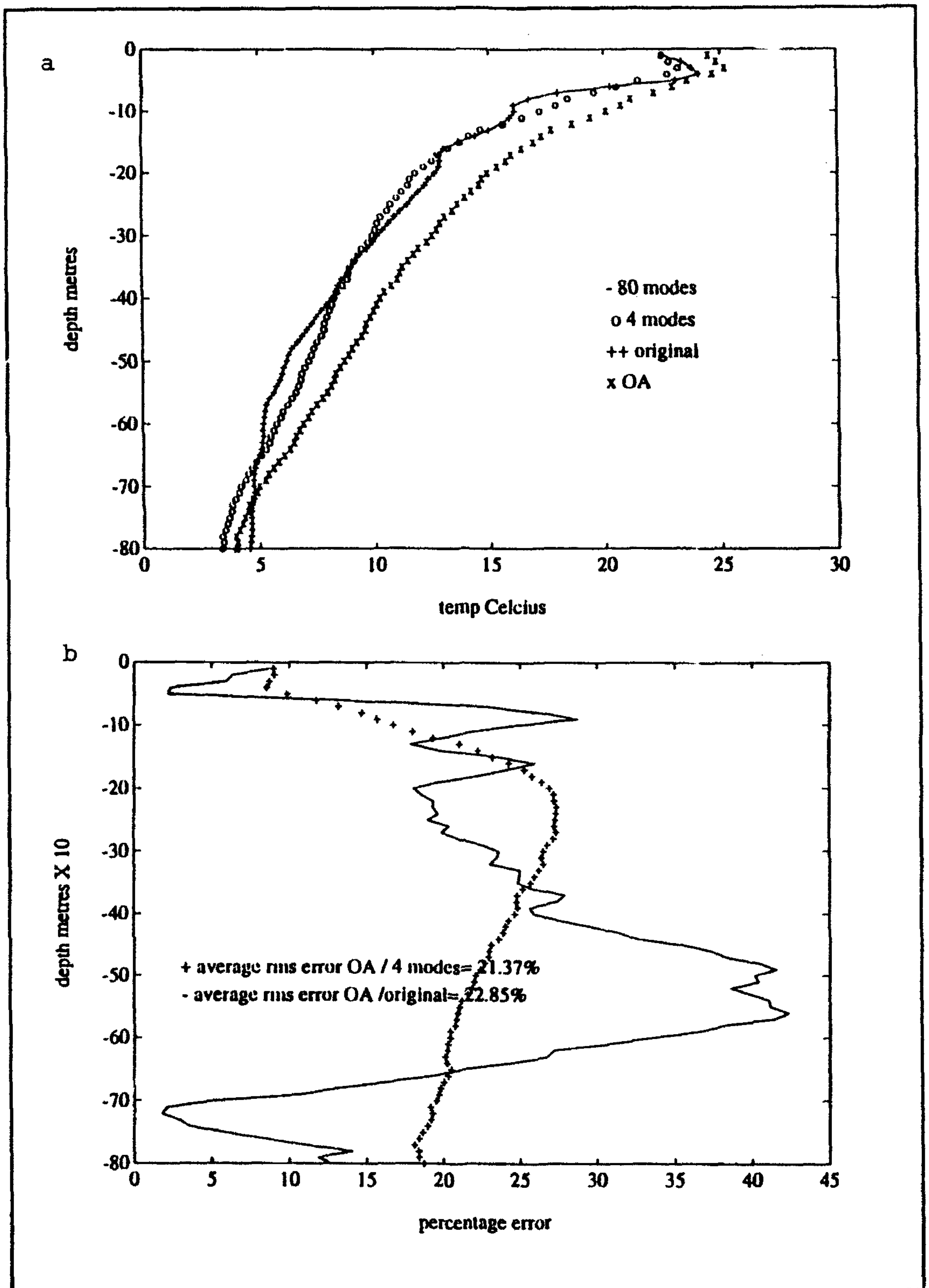

Figure 59 a, XBT 122 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

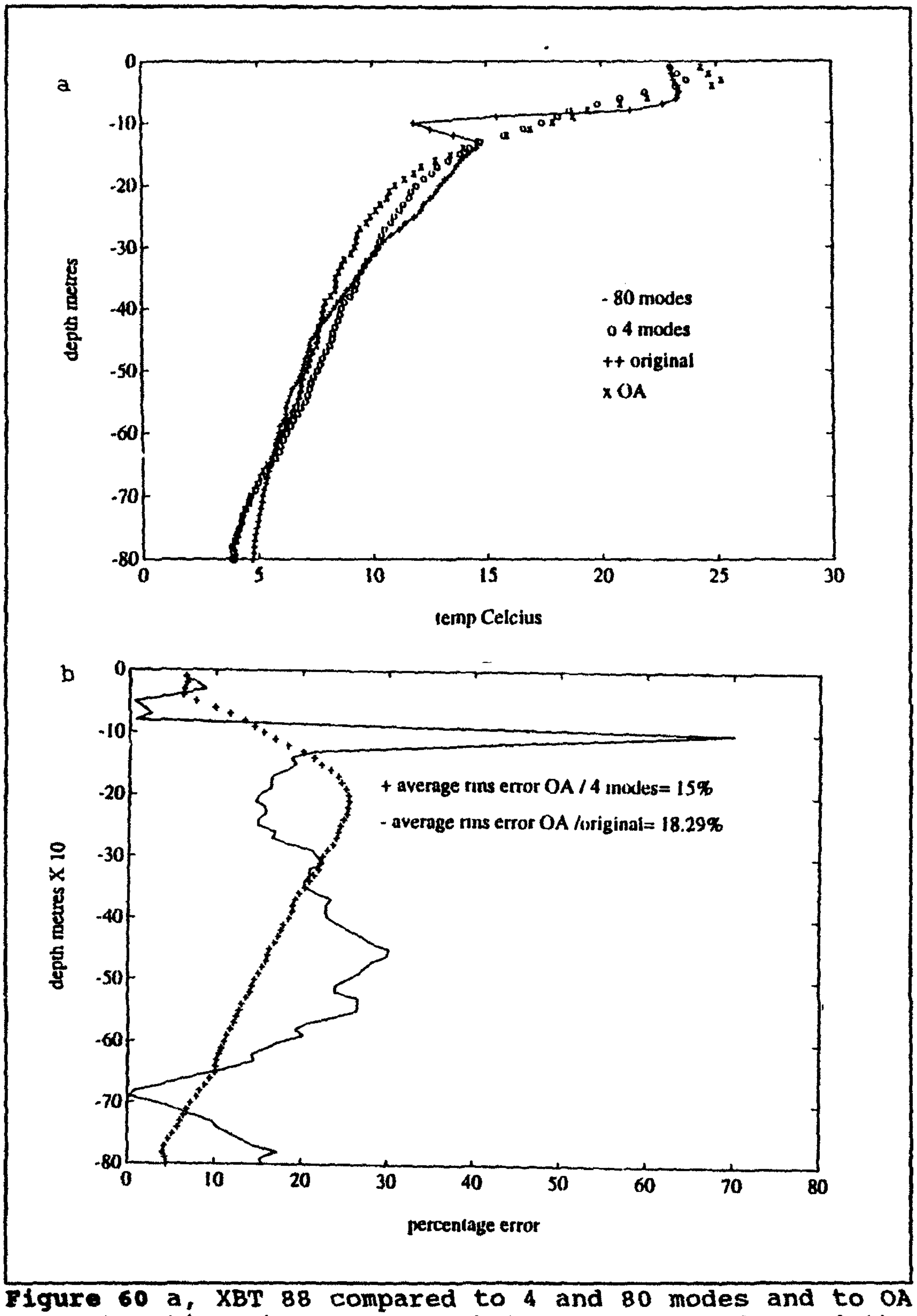

Figure 60 a, XBT 88 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

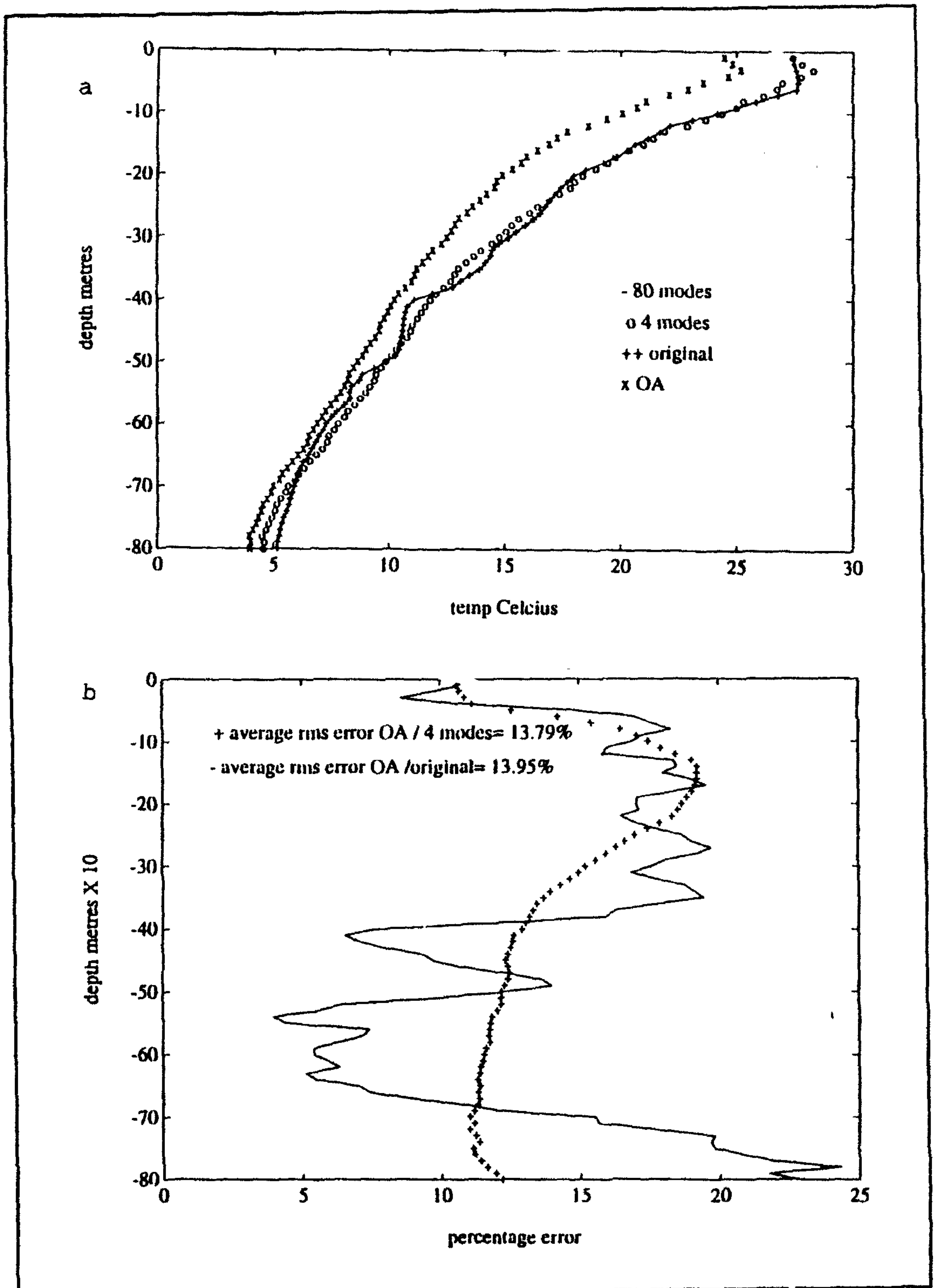

Figure 61 a, XBT 71 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

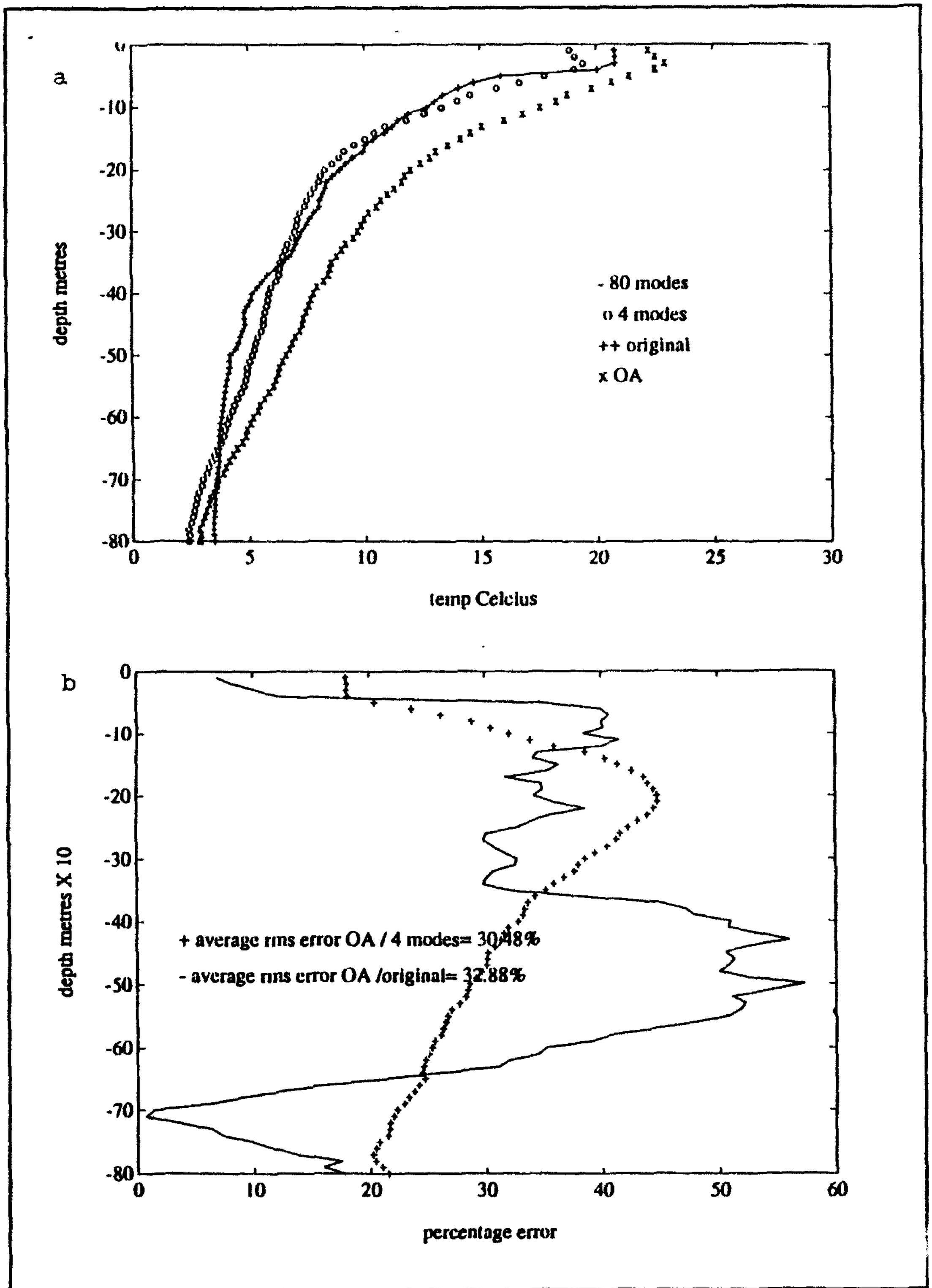

Figure 62 a, XBT 151 compared to 4 and 80 modes and to OA reconstruction; b, RMS error between OA, 4 modes and the original for each depth.

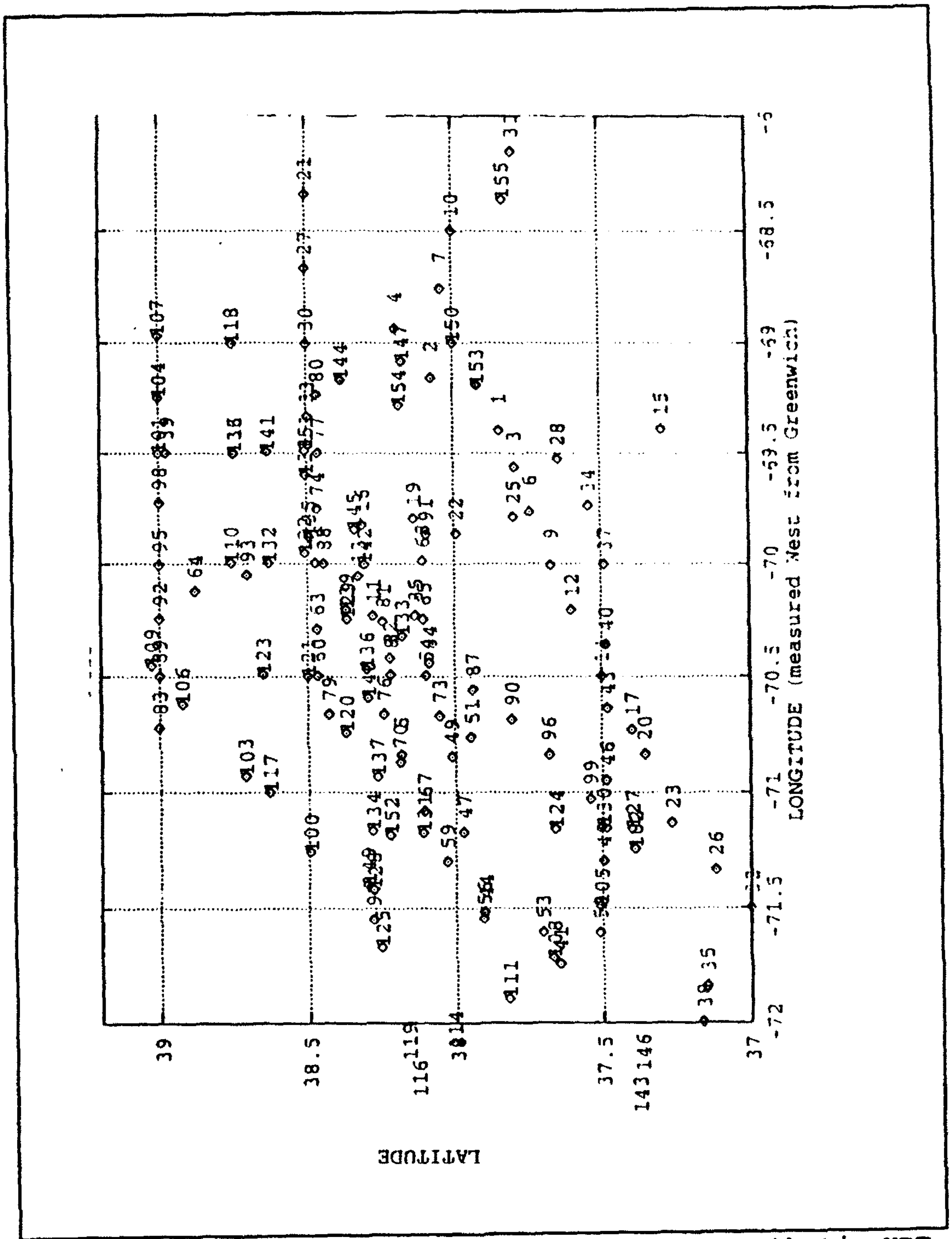

Figure 63 position of selected XBTs casts. Synthetic XBTs were calculated at grid points corresponding to original XBT sites.

VI THE RESULTS

A succession of error maps using reduced data are shown in Figures 64-70. It can be seen that the data were taken in two natural clusters. In the discussion that follows, only the Western cluster is considered. Within this cluster, most of the XBTs lie within an area covered by the 10% contour. After reducing the total number of XBTs to 40, the area covered by the 30% contour is still on the order of the size of the area covered by the original 10% contour.

The experiment was refined to identify a specific area that lay within the original 10% error contour line. Additionally, only XBTs taken in and around the designated area were included in the subsequent analysis. The cluster to the east was removed, plus a few XBTs laying in the extreme north of the analysis area. This resulted in 133 XBTs being used for the analysis (Figure 71). The aim of the experiment was to reduce the data set until the 30% contour intruded into the specified area. The sequence is shown in Figures 72 - 75. The 30% contour crosses the borders of the designated area when the data set is reduced to 69 XBTs.

It was concluded that, for a Gulf Stream meander, a minimum of 69 XBTs is required to adequately reproduce synthetic vertical temperature profiles with an acceptable error variance of 30%.

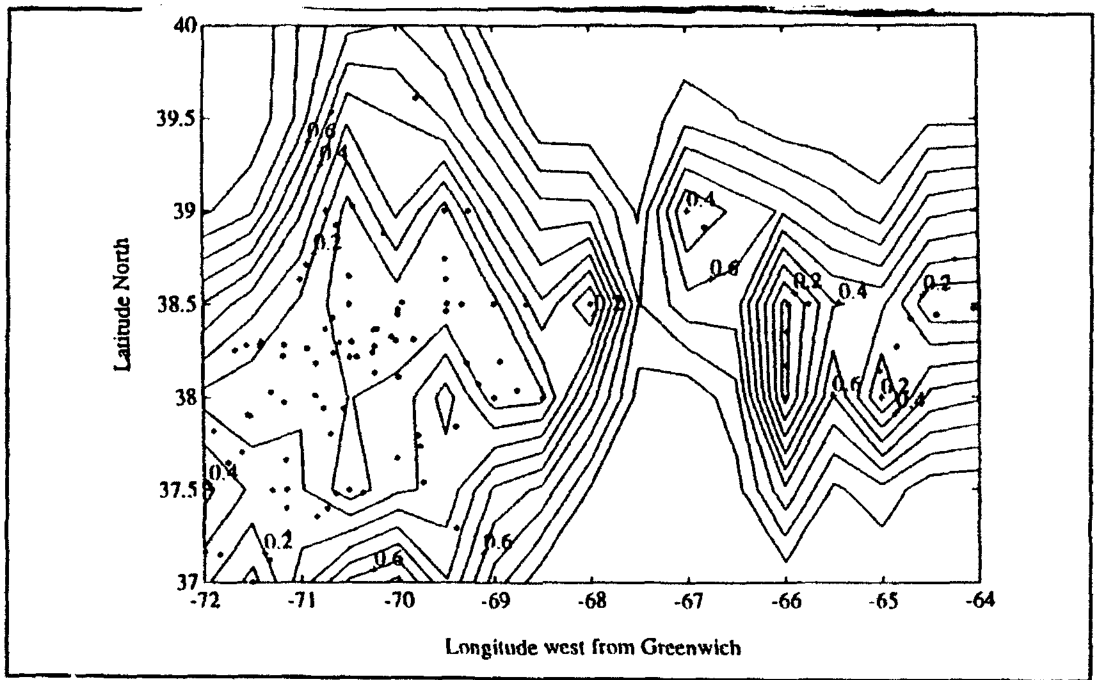

Figure 64 Reconstruction error variance using 100 out of 156 XBTs. The contour interval is 0.1

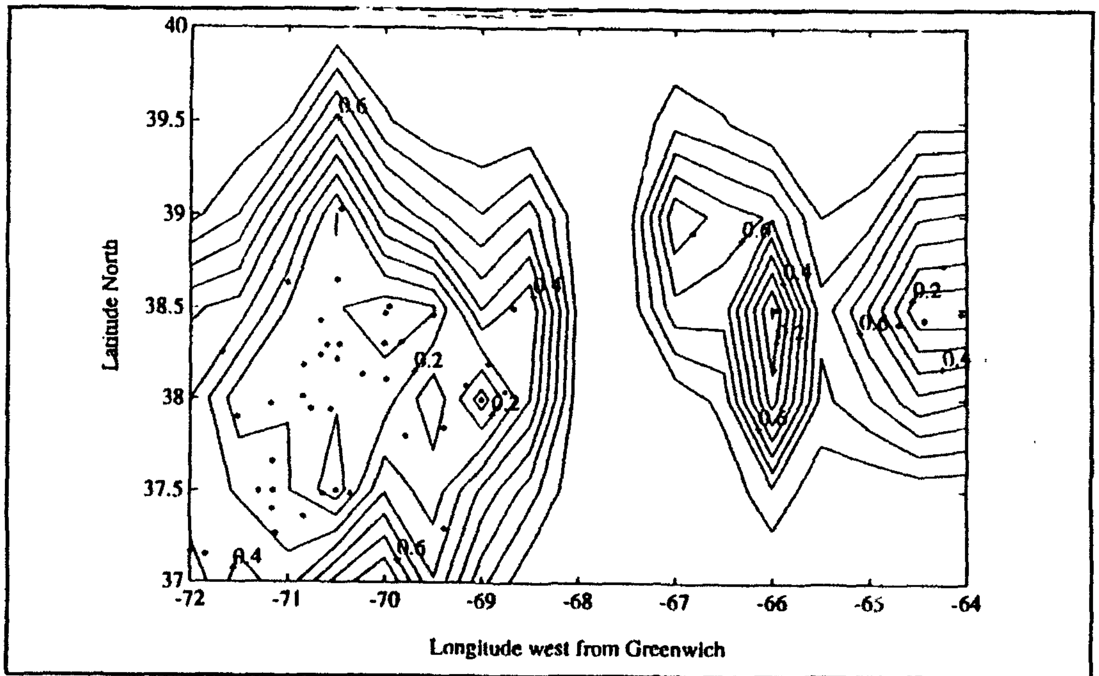

Figure 65 Reconstruction error variance using 50 out of 156 available XBTs.

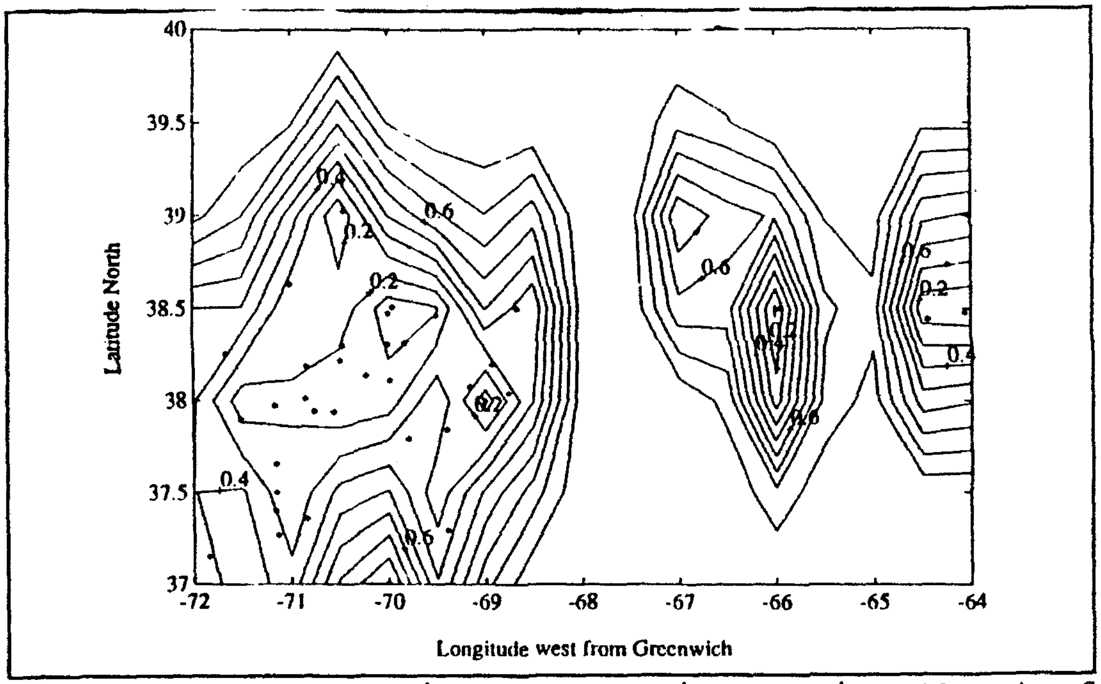

Figure 66 Reconstruction error variance using 40 out of 156 available XBTs.

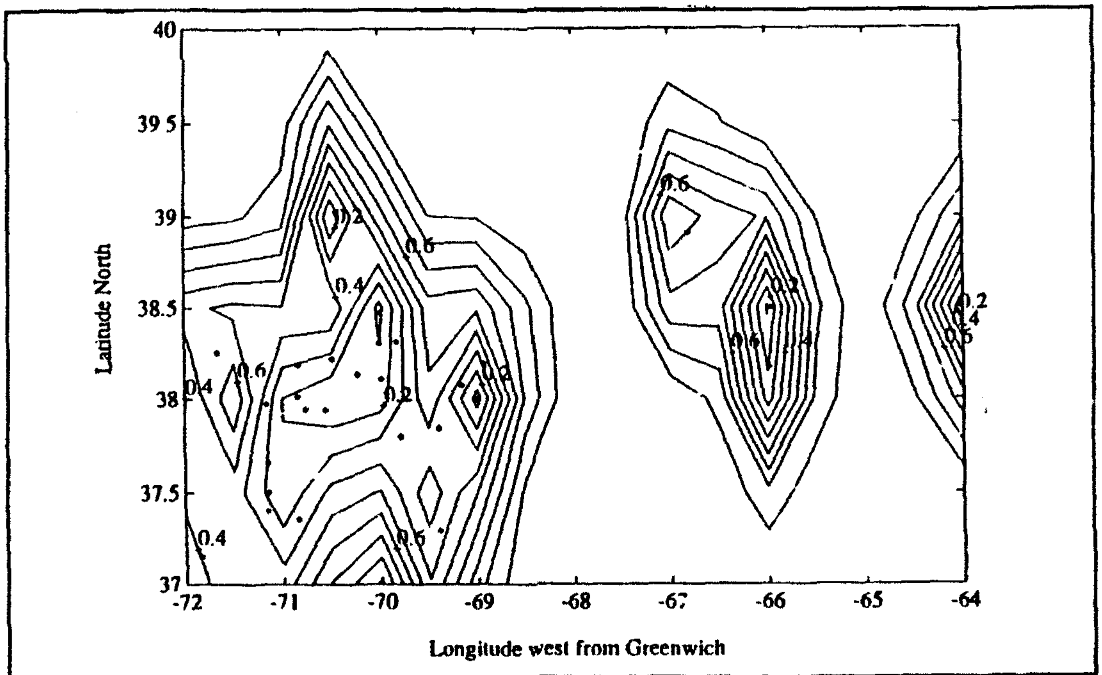

Figure 67 Reconstruction error variance using 30 out of 156 available XBTs.

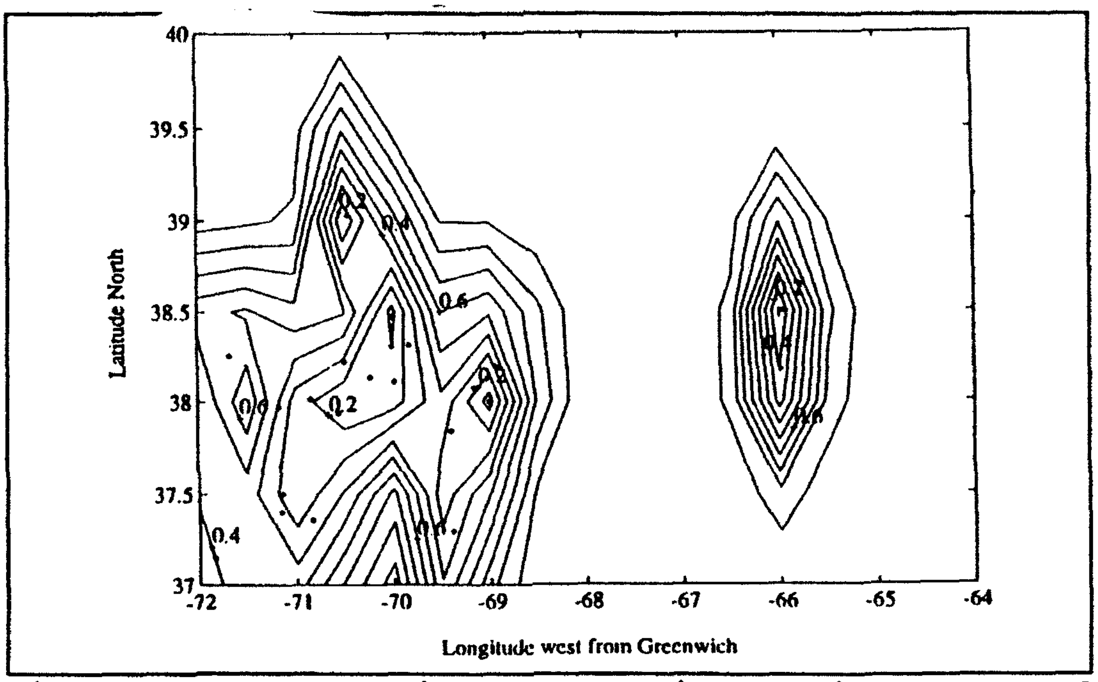

Figure 68 Reconstruction error variance using 25 out of 156 available XBTs.

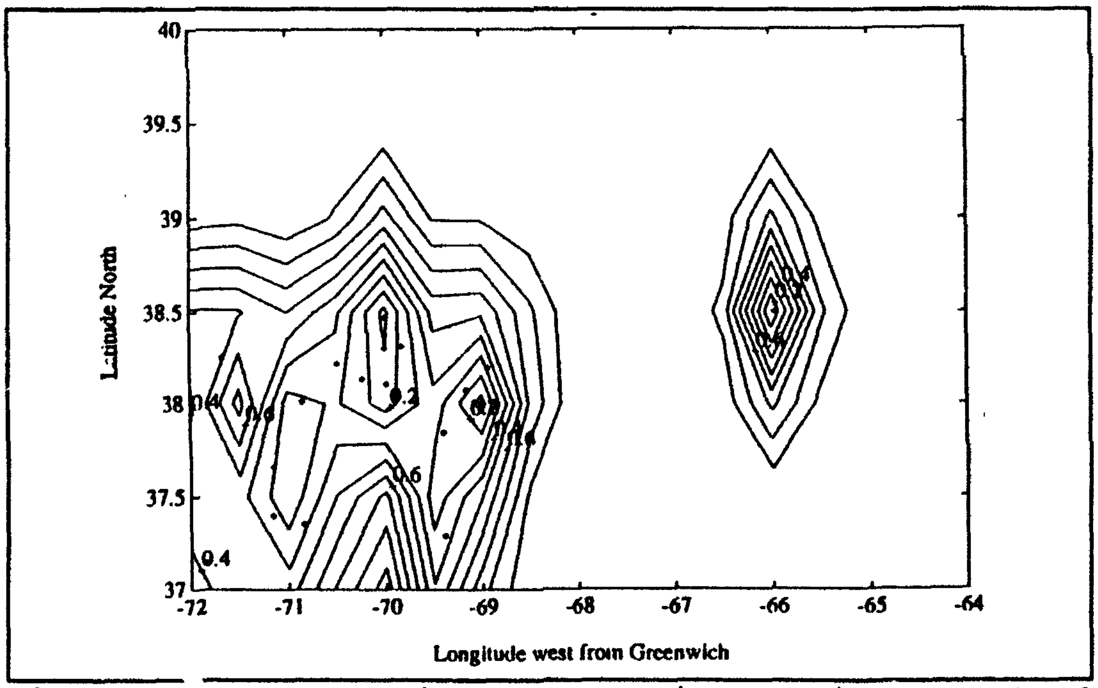

Figure 69 Reconstruction error variance using 20 out of 156 available XBTs.

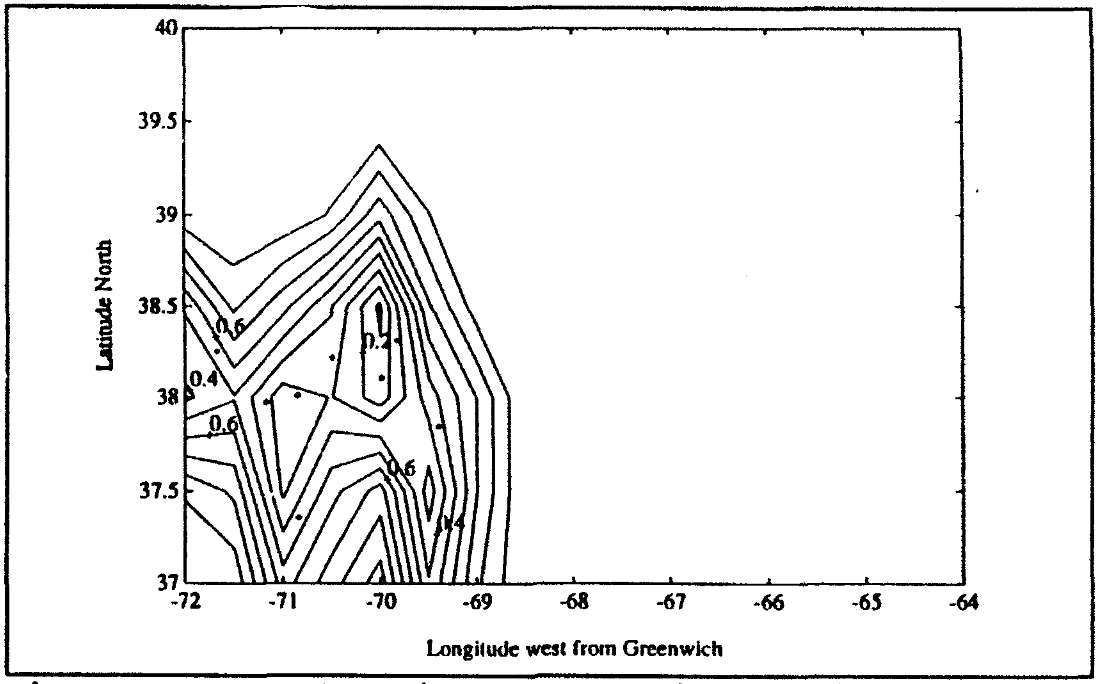

Figure 70 Reconstruction error variance using 10 out of 156 available XBTs.

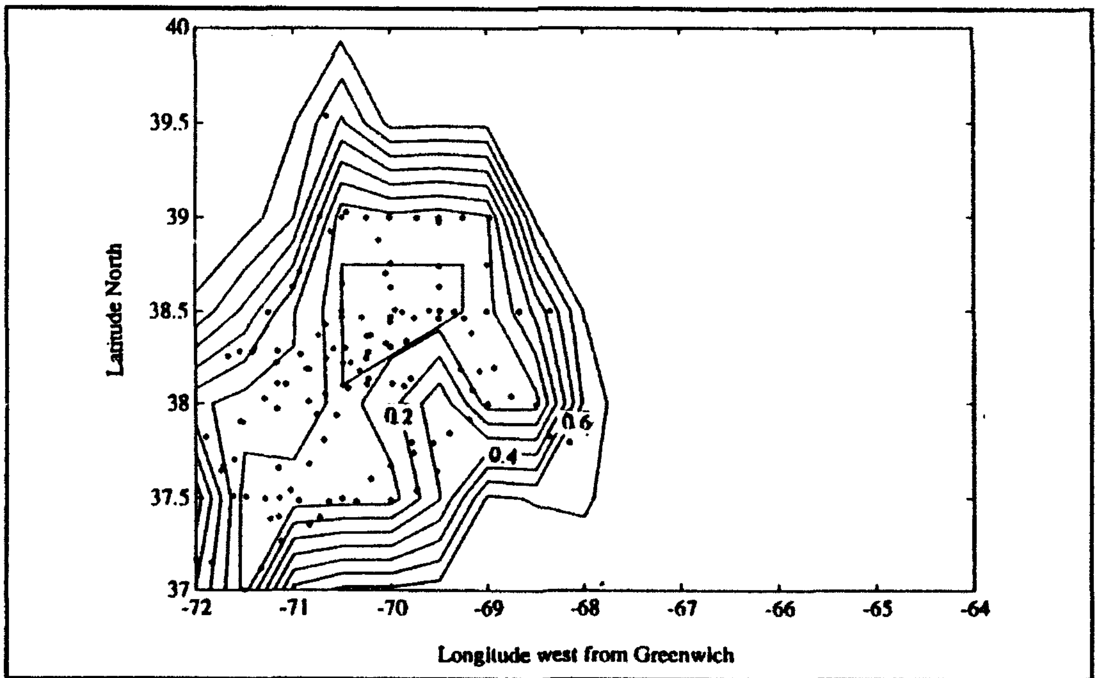

Figure 71 Reconstruction error variance using the 133 XBTs in the designated area.

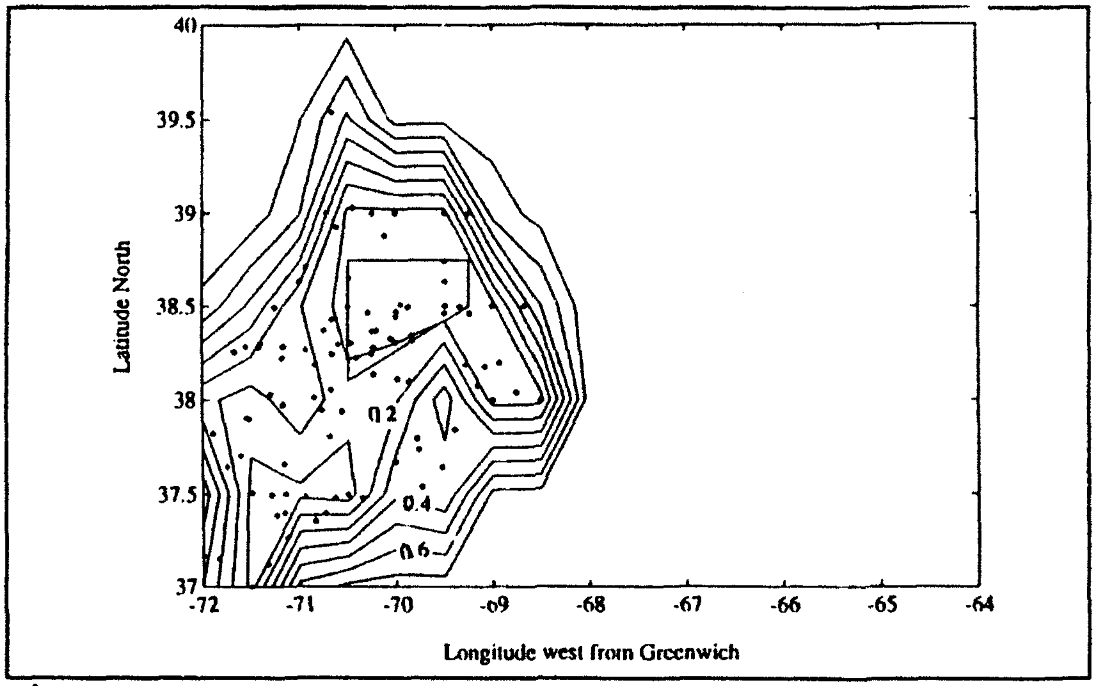

Figure 72
Reconstruction error variance using 100 out of the possible 133 XBTs.

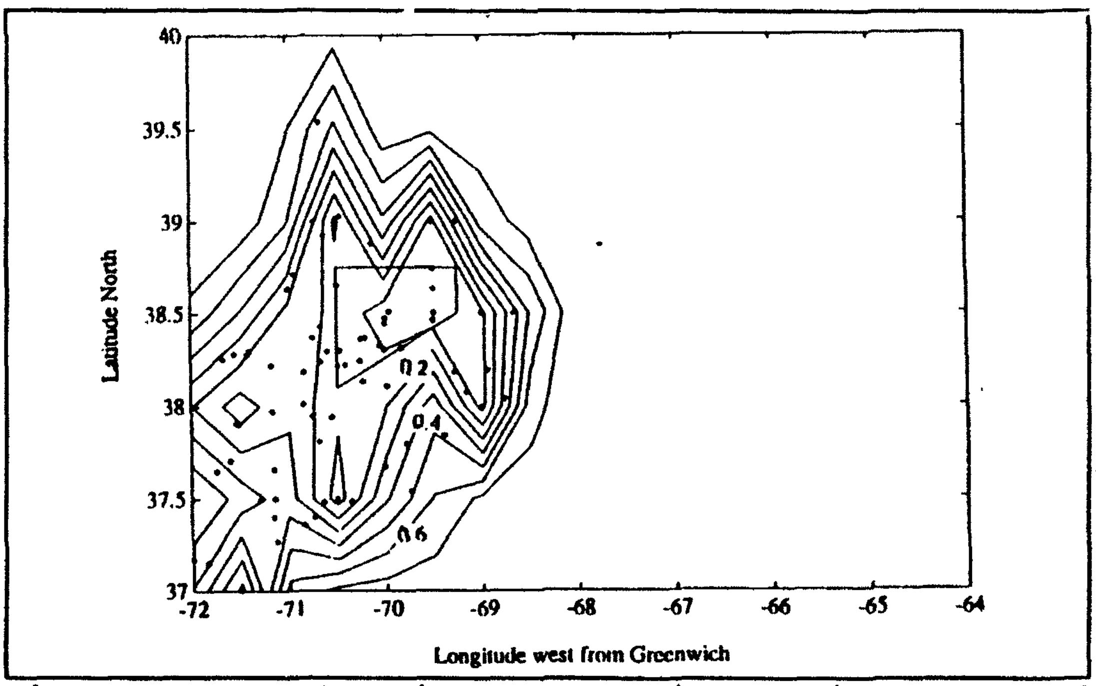

Figure 73 Reconstruction error variance using 75 out of the 133 available XBTs.

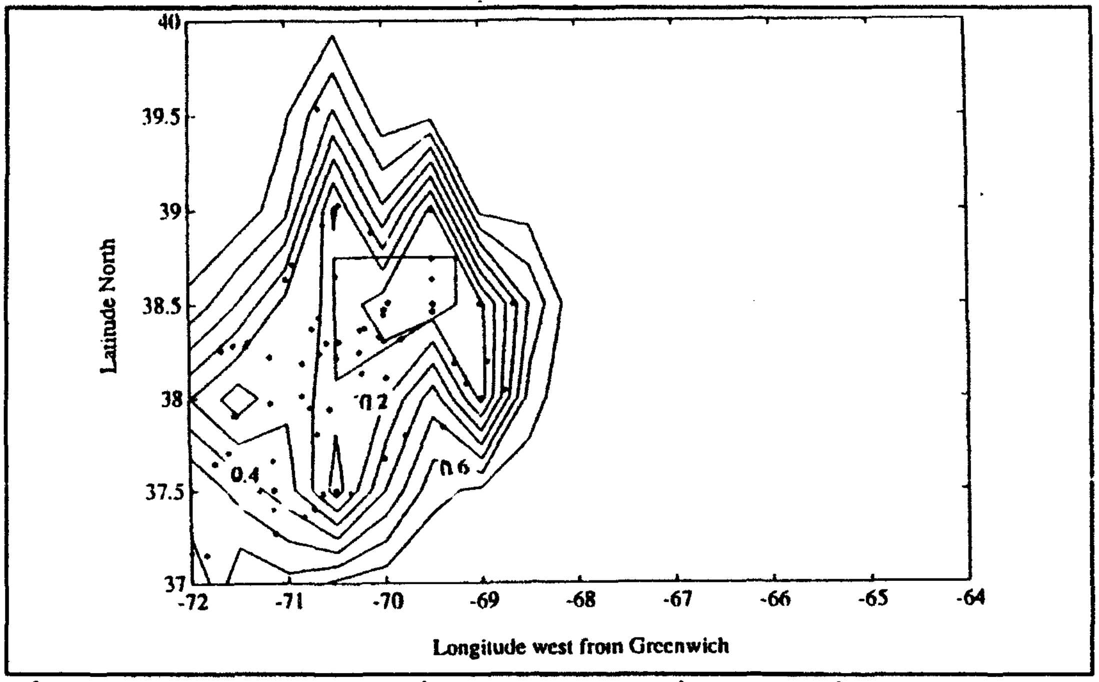

Figure 74 Reconstruction error variance using 70 out of the 133 available XBTs.

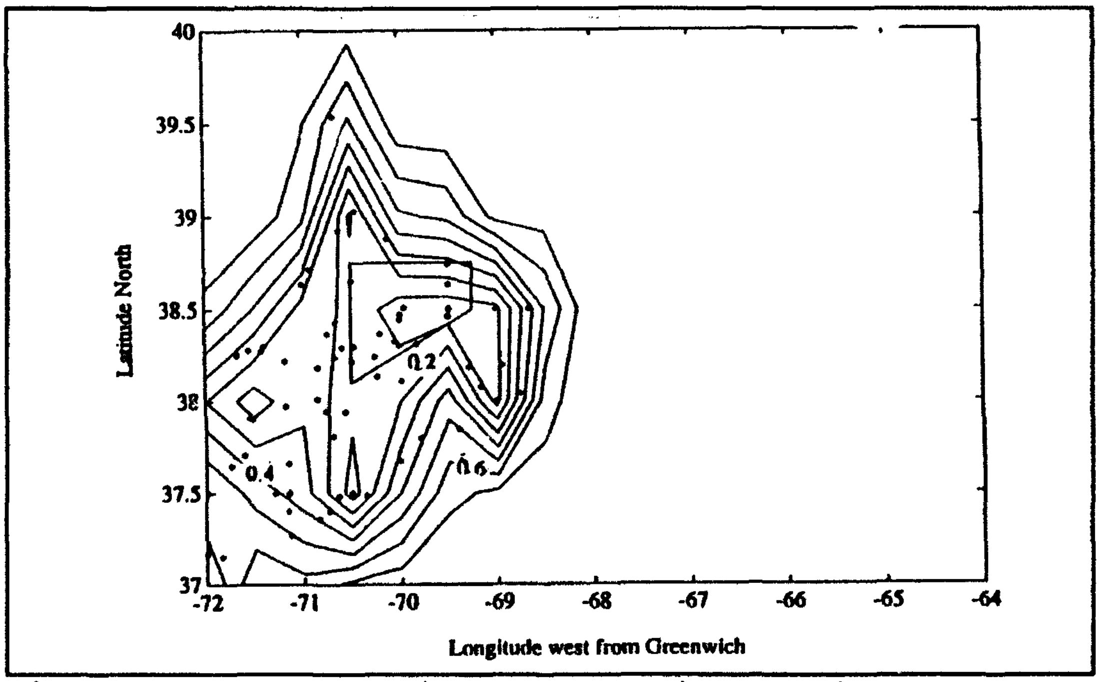

Figure 75 Reconstruction error variance using 68 out of 133 available XBTs. Note the 30% error variance line which crosses into the designated area.

VI DISCUSSION

A. THE RECONSTRUCTION

The difference between an original XBT profile and its reconstruction using only the first four EOFs has been of concern throughout this study. The analysis showed the average difference was of the order 6%. Thus, before the objective analysis is undertaken, a degree of error has already been introduced with the best that can be hoped for being a contour positioned on the error variance map accurate to within plus or minus 6 %. As a result, a synthetically reproduced XBT will have associated with it error due to the objective analysis and error due to the use of a truncated series of four EOFs. However, the first four modes account for over 98% of the variance, and reflect a minimum number that could reasonably be used. If higher accuracy was required, then more modes could have been considered but at the risk of of including noise from individual observations.

All these sources of variability are included in an 80 modes solution. The current situation is,in effect, a trade off; loosing some of the fine structure due to the small XBT data set and analysing only a limited number of vertical modes. Nevertheless, the study shows that a limited number of vertical modal amplitudes may be interpolated using objective

analysis to synthetically create XBTs at any given point within the region, with a definitive statement as to the level of confidence that can be placed in the reconstruction.

B. THE NUMBER OF XBTs

The last set of data runs in this study (Figures 71-75), provide an example of a realistic military or scientific scenario. The question asked was how many XBTs need to be taken in a Gulf Stream meander for a reasonable estimate of the ocean's vertical temperature structure can be inferred any where within the meander?

The area initially chosen was the area within the 10% error variance contour and reflects the area that was most heavily surveyed. The XBTs surrounding the area were also included, as they were considered to represent XBTs that would be dropped by units, whether by ship or aircraft, that were proceeding to or away from the area. Overall, XBTs are not dropped at regularly spaced intervals, and, although not random in nature, they tend to reflect a distribution that would be expected to be produced by several surface units attempting to track a covert submarine.

The area noted in Figures 71-75 is approximately 1400 square miles and was initially surveyed by 133 XBTs. The analysis indicates that, given a confidence level of 30% error, the same area could have been adequately sampled by 70

XBTs. This is a saving of nearly 50 percent in XBTs, but, more importantly, this study indicates that an effective analysis can be achieved in a complicated region with relatively few XBTs. Although this study has utilized data from within only one Gulf Stream meander, it provides a general indication of the amount of observations that would be needed within other Gulf Stream eddies or meanders.

C. OPTIMAL SPACING

The determination of the spacial correlation matrices resulted in parameter b, the e folding distance, to be defined and calculated for each of the modal amplitudes. This distance places a limit on the separation that can exist between two observations to be included in the analysis. From Table II it can be seen that the second modal amplitude gives the smallest value, a distance of 25 km. This value represents the maximum distance of separation that should exist between two adjacent observations. For the purpose of economy and military logistics the figure represents the optimal spacing that should exist between XBT cast sites.

The value of 25 km is approximately half that of the Rossby radius of deformation and is suggestive that a smaller grid scale would have been more appropriate.

D. RECOMMENDATIONS

To valididate the claim that 25 km is a good optimal distance, it would be of value to extend the study to consider regularly spaced XBTs (generating them synthetically, as the data from any real survey, by its very nature, will tend to have been erratically sampled data in terms of both time and space) with the distance between adjacent casts being gradually extended until the resulting error variance becomes unacceptable.

The current analysis also does not take into account the fact that each XBT was taken at different times. It was assumed throughout the study that all XBTs were valid at the analysis time. The study could be extended to take time into account, with the interpolation being adjusted to allow for an optimal value to be chosen both in terms of time and space (see Carter 1982).

A different correlation function could also have been fitted to the cross flow and along flow directions. This has value as it helps to account for the rapid changes that take place across the Gulf Stream front as opposed to the expected similarity in values taken along the front. In this study, the casts were taken within a well developed horseshoe shaped meander so it was decided to assume homogeneous statistics using the same correlation function in all directions.

However, the use of a non isotropic field should be considered.

A major extension to the study would be to obtain the principle modes by including data from other Gulf Stream eddies and meanders so as to build a climatology of Gulf Stream eddies. It is likely that the modal decomposition of a projection matrix defined from a larger data set would remove the spurious effects evident in the current study and allow for an improved reconstruction of the data when using the four principle modes. It is considered that a climatology of eddies rather than a climatology of the North West Atlantic would be of greater value in attempting to empirically model XBTs within the Gulf Stream region.

It is noted that the surface layer is poorly modelled, suggesting that two analyses may be required. One analysis for the surface layer, the upper 80 metres, and the second for the deeper water below the thermocline.

VII CONCLUSION

The creation of synthetic XBTs at regular locations within a Gulf Stream meander by the use of an objective analysis of modal amplitudes produced from the decomposition of the vertical temperature correlation projection matrix has been shown to be of value. Although there is a degree of error in the reconstruction, the value of the error is explicitly stated.

Using the error variance field generated from the objective analysis, it has been shown that within a 1400 square mile region of a warm Gulf Stream meander a minimum of 69 XBTs need to be taken in order for a synthetically produced XBT to be within 30% of its true value.

The spacial correlation statistics indicate that the optimal distance between XBT cast site must be 25 km or less.

LIST OF REFERENCES

Bretherton, F.P., R.E. Davis, and C.B. Fandry, 1976: A technique for objective analysis and design of oceanographic experiments applied to MODE-73. *DSR.*, 23:559-582.

Carnes, M , J. Mitchell, and P.W. Dewitt, 1990: Synthetic temperature profiles derived from geostat altimetry comparison with air dropped expendable bathythermograph profiles. *JGR., 17979-17991.*

Carter, E.F., 1983: The statistics and dynamics of ocean eddies. Ph.D. thesis, Harvard Univ, Cambridge, MA.

Carter, E.F., and A.R.Robinson, 1987: Analysis models for the estimation of oceanographic fields. *J.Atmos. Oceanic Technol.,* 4:49-74.

Clancy, M.R., and W.D. Sadler, 1992: The Fleet numerical Oceanographic suite of oceanic models and products. *J.Atmos.oceanic technol.*, 1: 0001-0021.

Clancy, M.R., P.A. Phoebus, K.D. Pollak, 1989: An optimal global scale ocean thermal analysis system. *J.Atmos. Oceanic technol.*, 7:7232-7254.

deWitt, P.W., 1987: Modal decomposition of the monthly Gulf Stream/Kuroshio temperature fields, NOO tech.Rep. 298, 40 pp, Nav. Oceanogr. Off., Stennis space Center, MS.

Dunteman, G., 1989: *Principle component analysis.* Sage, first edition, 96 pp.

Freeland, M.J., and W.J. Gould, 1976: Objective analysis of mesoscale ocean circulation features. *DSR.,* 23:915-923.

Gandin, L.S., 1965: *Objective analysis of meteorological fields*. Israel program for scientific translations, Jerusalem, 242 pp.

Harman, H., 1976: *Modern factor analysis*. University of Chicago press, chicago, 487 pp.

Hotelling, H., 1933: Analysis of a complex of statistical variables into principle components. *J. Ep.,* 24:417-441,498-520.

Hummon, J., T. Rossby, E. Carter, J. Lillibridge III, M.Liu, K.Schultz Tokos, S. Anderson-Fontana, A. Mariano, 1991: The anatomy of Gulf Stream meanders. Tech report ref 91-4, vol 1: Technical description and fall cruise data. Graduate school of Oceanography, Univ of Rhode Island.

Kidson, J.W., 1975: Eigenvector analysis of monthly mean surface data. *Mon. wea. Rev.*, 103:No3, 177-186.

Kundu, P.K., J.S. Allen, and R.L. Smith, 1975: Modal decomposition of the velocity field near the Oregon coast. *J.phys. Oceanogr.*, 5:683-704.

Kutzbach, J., 1967: Empirical eigenvectors of sea level pressure, surface temperature and precipitation complexes over North America. *J. Appl. Meteor.*, 6:791-802.

Lorenz, E., 1956: Empirical orthogonal functions and statistical weather prediction. Sci. Rep. No. 1. Statistical forecasting project, Dept of Meteorology, MIT, 49pp.

Mitchell, J.M., B. Dzerdzeevskii, H. Flohn, W.L. Hoime, H.H.Lamb, K.N. Rao, and C.C. WEallen, 1966: Climotological change. WMO, Tech Note no 79, WMO No 195 TP.100, 79pp.

Paegle, J.N., and R.B. Haslam, 1982: Empirical orthogonal functions estimates of local predictability. *J. Appl. Meteor.*, 21:117-126.

Pearson, K., 1901: On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, Ser 2, 6:559-572.

Preisendorfer,R.W., F.W. Zwiers, and T.P. Barnett, 1981: *Foundations of Principal Component Selection Rules*, Scripps Institution of Oceanography, University of California, pp192.

Robinson, A.R., and W.G. Leslie, 1985: Estimation and prediction of oceanic eddy fields. *Prg Oceanogr.*, 14:485-510.

Stidd, C.K., 1966: The use of eigenvectors for climatic estimates. *J. Appl. Meteor.*, 6: 255-264.

Tunnicliffe, P.A., and J.A. Cummings, 1991: A water mass empirical orthogonal function climatology for the Greenland-Iceland Norwegian Seas. Proc. MTS. 91 conf. New Orleans, *MTS.*, 708-712.

Wallace, J.M., and R.E. Dickinson, 1972: Empirical orthogonal representation of time series in the frequency domain. part I: Theoretical considerations. *J. Appl. Meteor.*, 11:887-892.

Watts, D.R., K.L. Tracey, and A.I. Friedlander, 1989: Producing accurate maps of the Gulf Stream thermal front using objective analysis. *JGR.*,94:8040-8052.

Wrigley, C., and J.O. Neuhaus, 1955: The use of electronic computers in principal axis factor analysis.

APPENDIX A.

Sep	17	88261:10:10	38	52.8	N	70	7.5	W	xbt	1
Sep	17	88261:11:14	38	42.3	N	70	3.1	W	xbt	2
Sep	17	88261:12:19	38	30.6	N	69	57.1	W	xbt	3
Sep	17	88261:19:26	37	49.5	N	68	21.8	W	xbt	4
Sep	17	88261:20:07	37	47.6	N	68	9.3	W	xbt	5
Sep	18	88262:14:11	38	0.0	N	66	0.0	W	xbt	6
Sep	18	88262:15:05	38	10.4	N	65	58.8	W	xbt	7
Sep	18	88262:16:07	38	21.4	N	65	58.5	W	xbt	8
Sep	18	88262:16:57	38	29.9	N	65	57.6	W	xbt	9
Sep	18	88262:17:40	38	30.0	N	65	45.1	W	xbt	10
Sep	18	88262:18:36	38	25.3	N	65	31.4	W	xbt	11
Sep	18	88262:19:34	38	17.4	N	65	16.7	W	xbt	12
Sep	18	88262:20:39	38	8.6	N	65	0.9	W	xbt	14
Sep	18	88262:21:38	38	16.3	N	64	50.8	W	xbt	15
Sep	18	88262:22:26	38	25.3	N	64	42.5	W	xbt	16
Sep	18	88262:23:19	38	26.6	N	64	26.4	W	xbt	17
Sep	19	88263:00:31	38	28.4	N	64	3.4	W	xbt	18
Sep	19	88263:02:52	38	44.3	N	64	14.6	W	xbt	19
Sep	19	88263:05:04	38	58.4	N	64	42.4	W	xbt	20
Sep	19	88263:07:29	39	0.1	N	65	15.2	W	xbt	21
Sep	19	88263:11:34	39	0.3	N	66	14.4	W	xbt	23
Sep	19	88263:14:06	38	54.8	N	66	48.6	W	xbt	24
Sep	19	88263:16:11	38	53.9	N	67	21.7	W	xbt	25
Sep	20	88264:04:20	38	14.7	N	70	15.5	W	xbt	26
Sep	20	88264:05:27	38	5.2	N	70	26.4	W	xbt	27
Sep	20	88264:06:25	37	56.4	N	70	33.5	W	xbt	28
Sep	20	88264:07:23	37	48.4	N	70	41.4	W	xbt	29
Sep	20	88264:08:22	37	40.9	N	70	50.5	W	xbt	30
Sep	20	88264:09:30	37	32.5	N	71	1.8	W	xbt	31
Sep	20	88264:10:44	37	23.4	N	71	14.4	W	xbt	33
Sep	20	88264:12:01	37	30.4	N	71	29.4	W	xbt	34
Sep	20	88264:13:18	37	40.1	N	71	43.1	W	xbt	35
Sep	20	88264:14:20	37	49.4	N	71	53.7	W	xbt	36
Sep	20	88264:15:32	38	0.6	N	72	6.6	W	xbt	37
Sep	20	88264:16:34	38	7.8	N	72	19.4	W	xbt	38
Sep	20	88264:20:34	38	9.8	N	72	9.8	W	xbt	39
Sep	21	88264:23:57	38	15.3	N	71	40.3	W	xbt	40
Sep	21	88265:03:52	38	16.9	N	71	25.1	W	xbt	41
Sep	21	88265:09:20	38	6.6	N	71	10.3	W	xbt	42
Sep	21	88265:10:25	38	17.0	N	71	9.8	W	xbt	43
Sep	21	88265:15:59	38	16.0	N	70	56.0	W	xbt	44
Sep	21	88265:21:39	38	17.8	N	70	35.7	W	xbt	45
Sep	23	88267:15:35	37	22.8	N	72	23.7	W	xbt	46
Sep	23	88267:17:22	37	22.3	N	72	12.7	W	xbt	47
Sep	24	88268:00:26	38	18.1	N	71	23.9	W	xbt	48
Sep	24	88268:02:54	38	13.3	N	71	10.9	W	xbt	49
Sep	24	88268:03:58	38	11.2	N	70	50.4	W	xbt	51
Sep	24	88268:05:03	38	13.1	N	70	29.5	W	xbt	52
Sep	24	88268:06:07	38	16.7	N	70	14.1	W	xbt	53
Sep	24	88268:06:58	38	19.7	N	70	3.3	W	xbt	54
Sep	24	88268:08:00	38	18.8	N	69	49.6	W	xbt	55
Sep	24	88268:09:01	38	8.2	N	69	47.8	W	xbt	56

Sep	24	88268:09:55	37	59.5	N	69	51.9	W	xbt	57
Sep	24	88268:10:56	37	47.8	N	69	47.3	W	xbt	58
Sep	24	88268:11:56	37	38.6	N	69	31.5	W	xbt	59
Sep	24	88268:13:48	37	32.4	N	69	44.3	W	xbt	60
Sep	24	88268:15:23	37	29.4	N	70	0.2	W	xbt	61
Sep	24	88268:17:12	37	29.1	N	70	21.6	W	xbt	62
Sep	24	88268:18:34	37	28.9	N	70	38.4	W	xbt	63
Sep	24	88268:19:59	37	29.2	N	70	56.9	W	xbt	64
Sep	24	88268:21:29	37	29.9	N	71	17.6	W	xbt	65
Sep	24	88268:23:00	37	30.7	N	71	36.6	W	xbt	66
Sep	25	88269:00:04	37	42.3	N	71	36.5	W	xbt	67
Sep	25	88269:00:59	37	54.4	N	71	32.8	W	xbt	68
Sep	25	88269:04:49	38	1.8	N	71	17.9	W	xbt	69
Sep	25	88269:08:33	38	6.5	N	71	4.8	W	xbt	71
Sep	25	88269:12:11	38	11.3	N	70	52.3	W	xbt	72
Sep	25	88269:13:26	38	3.2	N	70	40.5	W	xbt	73
Sep	25	88269:14:52	38	14.6	N	70	40.0	W	xbt	74
Sep	25	88269:15:53	38	25.8	N	70	40.0	W	xbt	75
Sep	25	88269:23:05	38	13.3	N	70	25.1	W	xbt	76
Sep	25	88269:23:58	38	8.1	N	70	14.0	W	xbt	77
Sep	26	88270:04:20	38	26.7	N	70	0.1	W	xbt	78
Sep	26	88270:07:35	38	5.7	N	69	52.1	W	xbt	79
Sep	27	88271:17:59	38	17.0	N	71	33.0	W	xbt	80
Sep	27	88271:19:27	38	29.8	N	71	15.3	W	xbt	81
Sep	27	88271:21:01	38	42.8	N	70	55.9	W	xbt	82
Sep	27	88271:22:35	38	55.5	N	70	37.2	W	xbt	83
Sep	27	88271:23:23	39	1.7	N	70	27.1	W	xbt	84
Sep	28	88272:00:36	39	15.7	N	70	31.4	W	xbt	85
Sep	28	88272:02:00	39	32.3	N	70	39.5	W	xbt	86
Sep	29	88273:11:21	38	38.0	N	71	0.1	W	xbt	87
Sep	29	88273:14:28	38	22.2	N	70	44.7	W	xbt	88
Sep	29	88273:17:32	38	39.1	N	70	29.4	W	xbt	89
Sep	29	88273:20:17	38	22.0	N	70	14.8	W	xbt	90
Sep	29	88273:23:09	38	37.9	N	69	59.9	W	xbt	92
Sep	30	88274:00:55	38	29.7	N	69	52.7	W	xbt	93
Sep	30	88274:04:11	38	30.3	N	69	36.0	W	xbt	94
Sep	30	88274:05:11	38	38.0	N	69	29.7	W	xbt	95
Sep	30	88274:08:01	38	23.0	N	69	10.0	W	xbt	96
Sep	30	88274:09:27	38	10.5	N	69	5.0	W	xbt	97
Sep	30	88274:10:23	38	0.0	N	69	0.0	W	xbt	98
Sep	30	88274:11:23	37	55.1	N	69	11.4	W	xbt	99
Sep	30	88274:12:27	37	50.6	N	69	23.8	W	xbt	100
Sep	30	88274:13:25	37	47.5	N	69	33.8	W	xbt	101
Sep	30	88274:14:24	37	44.3	N	69	46.0	W	xbt	102
Sep	30	88274:15:31	37	40.2	N	70	0.4	W	xbt	103
Sep	30	88274:16:33	37	36.1	N	70	12.5	W	xbt	104
Sep	30	88274:17:56	37	30.0	N	70	30.0	W	xbt	105
Sep	30	88274:19:21	37	23.9	N	70	44.0	W	xbt	106
Sep	30	88274:20:00	37	21.5	N	70	50.3	W	xbt	107
Sep	30	88274:21:47	37	16.1	N	71	7.8	W	xbt	108
Sep	30	88274:23:12	37	7.2	N	71	20.0	W	xbt	109
Oct	1	88275:00:27	37	0.0	N	71	30.0	W	xbt	110

Oct	1	88275:03:03	37	9.1	N	71	50.5	W	xbt	111
Oct	1	88275:03:57	37	10.0	N	71	59.9	W	xbt	112
Oct	1	88275:07:36	37	38.8	N	71	44.8	W	xbt	113
Oct	1	88275:08:57	37	54.1	N	71	31.1	W	xbt	114
Oct	1	88275:10:23	37	58.4	N	71	10.2	W	xbt	115
Oct	1	88275:11:27	38	0.8	N	70	51.0	W	xbt	117
Oct	1	88275:12:20	37	56.7	N	70	45.9	W	xbt	118
Oct	1	88275:17:41	38	5.9	N	70	29.8	W	xbt	119
Oct	1	88275:20:18	38	28.0	N	70	29.9	W	xbt	120
Oct	1	88275:21:14	38	28.1	N	70	17.6	W	xbt	121
Oct	2	88276:01:09	38	6.3	N	70	15.0	W	xbt	122
Oct	2	88276:02:13	38	6.5	N	69	59.3	W	xbt	123
Oct	2	88276:05:15	38	28.4	N	69	59.8	W	xbt	124
Oct	2	88276:06:09	38	27.9	N	69	45.2	W	xbt	125
Oct	2	88276:12:44	38	27.7	N	69	30.0	W	xbt	126
Oct	2	88276:13:54	38	27.9	N	69	14.3	W	xbt	127
Oct	4	88278:13:43	39	0.1	N	70	43.5	W	xbt	129
Oct	4	88278:15:06	39	0.0	N	70	30.0	W	xbt	130
Oct	4	88278:16:20	39	0.0	N	70	14.9	W	xbt	131
Oct	4	88278:17:31	39	0.1	N	70	0.5	W	xbt	132
Oct	4	88278:18:52	38	59.9	N	69	43.6	W	xbt	133
Oct	4	88278:19:55	39	0.0	N	69	30.1	W	xbt	134
Oct	4	88278:21:06	39	0.0	N	69	15.0	W	xbt	135
Oct	4	88278:22:28	39	0.0	N	68	58.1	W	xbt	137
Oct	5	88279:05:54	38	45.4	N	69	59.7	W	xbt	138
Oct	5	88279:11:21	38	44.7	N	69	30.0	W	xbt	139
Oct	5	88279:15:25	38	45.0	N	69	0.1	W	xbt	140
Oct	6	88280:13:12	38	30.0	N	70	30.0	W	xbt	141
Oct	6	88280:19:23	37	39.6	N	71	9.3	W	xbt	142
Oct	6	88280:21:07	37	24.1	N	71	9.3	W	xbt	143
Oct	6	88280:22:36	37	29.9	N	71	8.6	W	xbt	144
Oct	7	88281:05:37	38	10.8	N	70	19.3	W	xbt	145
Oct	7	88281:07:34	38	18.1	N	70	28.1	W	xbt	146
Oct	7	88281:08:42	38	22.1	N	70	12.3	W	xbt	147
Oct	7	88281:11:06	38	18.4	N	70	0.3	W	xbt	148
Oct	7	88281:11:42	38	20.4	N	69	50.6	W	xbt	149
Oct	9	88283:11:44	38	30.3	N	69	29.3	W	xbt	150
Oct	9	88283:13:43	38	11.1	N	69	17.1	W	xbt	151
Oct	9	88283:14:26	38	4.5	N	69	9.4	W	xbt	152
Oct	9	88283:15:25	38	11.8	N	68	58.4	W	xbt	153
Oct	9	88283:15:35	38	11.8	N	68	56.1	W	xbt	153
Oct	9	88283:16:40	38	2.4	N	68	45.4	W	xbt	154
Oct	9	88283:17:40	38	0.0	N	68	30.0	W	xbt	155
Oct	9	88283:19:00	37	50.0	N	68	15.0	W	xbt	156
Oct	10	88284:15:39	37	17.6	N	69	23.5	W	xbt	157
Oct	12	88286:11:11	38	30.0	N	68	0.1	W	xbt	158
Oct	12	88286:12:35	38	30.0	N	68	20.6	W	xbt	159
Oct	12	88286:13:51	38	30.1	N	68	40.0	W	xbt	160
Oct	12	88286:15:11	38	30.0	N	69	0.3	W	xbt	161
Oct	12	88286:16:28	38	29.8	N	69	20.2	W	xbt	162
Oct	12	88286:19:35	38	44.7	N	69	30.0	W	xbt	163
Oct	12	88286:20:46	38	58.4	N	69	29.9	W	xbt	164

Oct	12	88286:22:04	39	16.2	N	69	30.6	W	xbt 165
Oct	13	88287:00:05	39	36.5	N	69	47.7	W	xbt 166

```
c           APPENDIX 1B

            program loadbathys
c
c
c*************************************************************************
c this file  loads the bathys into an array. and calculates correlation.
c*************************************************************************

          integer m,n,p,iy,iz,q
           real sumab,corro,a,b,y,z ,tempvar(80,156),sumvar(80)
           integer counta(80,156),count
           real deptha(80,156),corrol(80,82)
           real tempa(80,156),mean(82),times,volt,qual

           real sumsqa,sumsqb,depthc(2000,156),tempc(2000,156)
           character lat*10,long*12,time*4
           character recnuma(80,156)*8,renum(156)*9
           integer yearday


c ***************************************************************


c LOAD IN DATA


              write(*,*) 'loading bathys'
              open(unit = 4, file = 'deepname',status = 'old')
              open(unit = 20, file = 'name')
              open(unit = 21, file = 'namepos')
              p = 0

              do 200 n = 1,156
                   read(4,'(a9)',end = 230) renum(n)
                   p = p + 1
                   write(20,*) p,'     ',renum(n)
                   open(unit= 3,file ='/usr/whitney_dl/xbt/'//renum(n))
c
                   rewind 3
                   read(3,*)
                   read(3,210) yearday,time
                   read(3,220) lat,long
                   read(3,*)
                   read(3,*)
c                      write(*,*) yearday,time
                     write(*,*) lat,long
                     write(21,*) lat,long
210                format(9x,i3,t29,a6)
220                format(6x,a10,t25,a12)
c
                   do 240 m = 1,2000
                          read(3,*,end = 200) times,depthc(m,n),
```

```
      $                         volt,tempc(m,n),qual
240                   continue
                      write(*,*) p
200             continue

230             close(4)
                close(3)
                write(*,*) p,'  bathys loaded'
c

*************************************************************
c  This section calls redata and calculates temp at
c 10m increment depths, starting at 5m, for each bathy and loads
c them into arrays. Also finds mean temp for given depth taken over
c all bathys.
c*************************************************************
c  CALCULATE SMOOTHED TEMP AND MEAN FOR GIVEN DEPTH
c
                open(unit =12 ,file = 'meantemp')

                 do 40 y = 5,800,10
                      sum = 0
                      iy = 1+((y-5)/10)
                      call redata(y,iy,renum,count,counta,recnuma,deptha,
      $               tempa,depthc,tempc,p)
c
                      do 30 n = 1,p
                          sum = sum + tempa(iy,n)
30                    continue
                      mean(iy) = sum/count
                     write(12,*) y,mean(iy)
40           continue


c*************************************************************
c SEND TO OUTPUT
c
                  open(unit = 13 , file = 'profile.mat')

                   do 100 n = 1,p
                      do 110  iy = 1,80
                      write(13,*) tempa(iy,n)
110                     continue
100                continue

c now loop through each depth calculating the corrolation compared
c with the shallow depth.
c*************************************************************
c
c  CALCULATE CORRALATION

ccc            open(unit = 10,file = 'lbokat')
            open(unit = 8,file = 'corrl.mat')
```

```
        do 300 y = 5,800,10          ! the "shallow" depth ( the m loop)
              iy = 1 + (y-5)/10
               write(*,*) 'depth ', y ,iy
c
               do 305  z  = 5,800,10   ! the "deep" depth (the  n loop)
                                       ! compare each deep with shallow
c                   set constants, counter etc to zero
                    a = 0
                    b = 0
                    sumab = 0
                    sumsqa = 0
                    sumsqb = 0
                    corro = 0
                    iz = 1+(z-5)/10

                    do 310 n =1,p        ! loop through each "deep"
                       m = n               ! record
                       q = n
                     if ((deptha(iy,m)).eq. (0.0)) goto 310
                     if((deptha(iz,n)).eq.(0.0)) goto 310

500                    if (recnuma(iy,m).eq.recnuma(iz,n)) then
cccc    write(10,*) m,iy,deptha(iy,n),tempa(iy,m),recnuma(iy,m)
cccc    write(10,*) n,iz,deptha(iz,n),tempa(iz,m),recnuma(iz,n)
                          a = tempa(iy,m) - mean(iy)
                          sumsqa = sumsqa + a**2
                          b = tempa(iz,n) - mean(iz)
                          sumsqb = sumsqb + b**2
                          sumab = sumab +(a*b)

                          else
                          do 510 m =1,p
                     if (recnuma(iy,m).eq.recnuma(iz,n)) goto 500
510                       continue
                          goto 310

                      end if
c
310                 continue               ! with next record
                    corro = sumab/sqrt(sumsqa *sumsqb)
                    corrol(iy,iz) = corro
                    write(8,*) corrol(iy,iz)

305            continue                    ! next "deep" depth
c
300     continue                           ! next "shallow" depth
c
        end
```

```
c     APPENDIX 2B


      subroutine redata(z,iy,renum,count,counta,recnuma,deptha,
     $                tempa,depthc,tempc,p)
c******************************************************************
c this program finds temp for a given depth for all records
c and stores the value of temp and depth in arrays and passes them
C back to vertcorro.f . The depth for given temp is found by
c  averaging over a 10m bin. The value of depth used is passed in
c the call from main program.
c******************************************************************

       parameter (mm =2)
       real t,depth(mm),temp(mm),z,tempc(2000,156)
       real deptha(80,156),tempa(80,156),depthc(2000,156)
       character renum(156)*8,recnuma(80,156)*8
       integer n,count,counta(80,156),iy
       integer counter,nodatapt,p
       real add
c*****************************************************************
c        open(unit = 10,file = 'datapt') ! testing for data
c
c SET COUNT  the number of records processed  for a given depth
c
              count = 0
c
1010       do 600 n = 1,p ! loop through each bathy

                    add = 0       ! the sum of data points us d in a bin
                    counter = 0   !a count of number of data points
                                  ! used in a bin
c
                do 2000 m = 1,2000       ! loop through all data points
c
                    depth(1) = depthc(m,n)
                    temp(1) = tempc(m,n)
c  SELECT DATA POINTS IN BIN
                    if (depth(1).ge.z-5.and.depth(1).le.z+5) then
                       counter = counter +1   ! add up number of
c                                               data points
                       add = add + temp(1)    ! sum values of
c                                             ! data points
                       t = add/counter        ! mean temp for
c                                             ! given depth for
c                                             ! given bathy

                       if (depthc(m+1,n).gt.z+5) goto 5000
                        if ((tempc(m+1,n)).le.(0.0)) goto 5000
                       goto 2000
                       else
                       goto 2000
C
c LOAD VALUES INTO ARRAYS FOR PASSING BACK.
```

```
c
5000                    count = count + 1       ! increment counter
                        counta(i  n) = count
                        tempa(iy,n) = t    ! value of t placed in array
                        recnuma(iy,n)= renum(n) ! name of record beng read
                        deptha(iy,n) = z
                        nodatapt = counter*count  ! calculate number of
c                                                    ! data points

                        goto 600
                      end if
2000              continue

600             continue
c                write(10,*)z,nodatapt ,count,counter ! sends to file
                END
```

```
c      APPENDIX 3B
c
      program distcorro


        parameter(max = 56)
        implicit real*8 (a-z)
        real*8 dpr /57.29577951308232/
        real*4 dist(max,max)
        real declat,declong,lat(max),long(max),maxvalue,noofbins
        integer n,m,z,lati,longi,p,width
        real mode(4,max),corr(4,max),value(max,max)
        integer inoofbins

c ************************************************************

c This part calculates the distance between any two points and stores in
c an array


        open(unit = 3,file = 'decpos.mat')
        open(unit = 1, file = 'deeppos')

        do 10 n = 1,max
            read(1,20) lati, declat,longi,declong
20          format(x,i2,x,f5.2,4x,i3,x,f5.2)
            declat =declat/60
            declong = declong /60
            lat(n) = lati + declat
            long(n)=  longi - declong
            write(3,*) long(n),lat(n)
10      continue
c
         do 30 n = 1,max
            do 40 m =1,max
c
               t = (long(n) - long(m))/dpr
               dsinr= dsin(t) * dcos(lat(m)/dpr)
               r = dasin(dsinr)
               dsins = dsin(lat(m)/dpr)/dcos(r)
               s = dasin(dsins)
               dcosd = dcos(r) * dcos(s - (lat(n) /dpr))
               d = dacos(dcosd) * dpr
c
               dist(n,m) = d * 111.6114
c
40          continue
30       continue
c
c *****************************************************************

c  Now load the modal amplitudes. (just the principle ones for now)

          open(unit = 2, file = 'single.mat')
```

```
                do 110 z = 1,4
                do 100 n = 1,max
                   read(2,*) mode(z,n)

100             continue
110             continue
c
c  **************************************************
c
c  find max distance   for this data set
c
          maxvalue = 0.0
            do 210 n =1,max
            do 200 m = 1,max
              if( dist(n,m).gt. maxvalue) maxvalue = dist(n,m)
200         continue
210         continue
          write(*,*) 'maxvalue = ', maxvalue

c*****************************************************************
c
c    create bins by distance .
c
              write(*,*) 'enter bin width'
              read(*,*) width
              end = max * max
              noofbins = maxvalue/width
              inoofbins = int(noofbins)


C ***************************************************************
c
c  Calculate corrolation as function of distance,for each bin(p).


                  open( unit = 10,file ='smith.1')
                  open( unit = 11,file ='smith.2')
                  open( unit = 12,file ='smith.3')
                  open( unit = 13,file ='smith.4')

                  do 700 z  = 1,4
                  do 600 p = 1,inoofbins

                     call calc(mode,z,p,corr,width,dist)

600                continue
c
c             *************************
c
c   send corrolation for each bin to file smith
c
                   do 400 p = 1,inoofbins
                    write(9 + z,*) p*width,corr(z,p)
```

```
400                 continue
700                 continue
c
c*********************************************************************************
c
          end
```

```
c
            subroutine calc(mode,z,p,corr,width,dist)
c
          parameter(max = 56)
          real modea,modeb,meanmodea,corr(4,max)
          real meanmodeb,summodea,summodeb,sumab,sumsqa,sumsqb
          real mode(4,max),dist(max,max)
          integer p,counter,n,m,a,b,z,width,c,numberb(5000)
          integer numbera(5000)
c
c ***************************************************************
c
c           open( unit = 1, file = 'mean')
c           open(unit = 2, file = 'look')
c           open(unit = 3, file = 'aloop')
c           open(unit = 4, file = 'bloop')

c   set counter and all variables back to zero for a new p (bin).
c
                  counter = 0
                  modea = 0
                  modeb = 0
                  meanmodea = 0
                  meanmodeb = 0
                  summodea  = 0
                  summodeb  = 0
                  sumab =  0
                  sumsqa = 0
                  sumsqb = 0
                  corr(z,p) =  0

c
c*********************************************************************
c   Determine which data points are used for a given bin.


                     do 300 n =1,max
                     do 310 m = n,max
                           a = p*width
                           b = p*width -25
c                           write(*,*) n,m
         if (dist(n,m).lt.a.and.dist(n,m).ge.b) then

                                   counter = counter +1
                                   c = counter
                                    write(*,*) a,c
                                   numbera(c) = n
                                   numberb(c) = m

                                  end if
310                  continue
300                  continue

c           write(*,*) p,p*width,c
```

```
c the value c is the number of data point pairs in a bin.
c numbera is the array number containing the first point and
c numberb is the array number containing the second point.
c
c **********************************************************
c now add up value of all data points used for given bin.and
c  calculate the mean for "a" loop.

                  counter = 0

                  do 400 n = 1,max
                  do 410 m = 1,c
                        if(n.eq.numbera(m)) then
                        summodea = summodea + mode(z,n)
c                        write(3,*) p,n,m,mode(z,n)
                         counter = counter + 1
                        goto 400
                        else
                        goto 410
                        end if
410                continue
400                continue
                 meanmodea = summodea/counter

c*********************************************************
c now do same for "b" loop.

                  counter = 0
                  do 500 n = 1,max
                  do 510 m = 1,c
                        if(n.eq.numberb(m)) then
                        summodeb = summodeb + mode(z,n)
                         counter = counter + 1
c                        write(4,*) p,n,m,mode(m,n)
                        goto 500
                        else
                        goto 510
                        end if
510                continue
500                continue
               meanmodeb = summodeb/counter



c              write(1,*) z,p,meanmodea,meanmodeb

c***********************************************************************
c
c     Calculate components for corrolation
c
                  do 320 n   = 1,max
                  do 330 m   = n,max
                        a = p*width
```

```
                     b = p*width -25
        if (dist(n,m).lt.a.and.dist(n,m).ge.b) then

                     modea  = mode(z,n)   - meanmodea
                     modeb  = mode(z,m)   - meanmodeb
                     sumab  = sumab   + modea*modeb
                     sumsqa = sumsqa + modea**2
                     sumsqb = sumsqb + modeb**2

                      end if
330              continue
320              continue
c
c  calculate correlation, for passing back.
c
                 corr(z,p) = (sumab)/sqrt(sumsqa*sumsqb)
c                 write(*,*) z,p,corr(z,p)

                 end
```

```
c       APPENDIX 4B

c
c This will be a program that determines correlation
c  for any distance by fitting to the data in 'smith'.
c
c*************************************************************
      parameter(i = 8)
      integer m,j,n,count,istep
      real  cr(4,i),cn(4,28),error(4,i),sumerror,d(4,28),olderror
      real a,b,newerror,deltaa,deltab,deltaea1,deltaea2,r
      real deltaeb1,deltaeb2,incrementa,incrementb
      real suma,sumb,deda,dedb
      real alfa(4),beta(4)
c *************************************************************

             open( unit = 11, file = 'smith.1')
             open( unit = 12, file = 'smith.2')
             open( unit = 13, file = 'smith.3')
             open( unit = 14, file = 'smith.4')
             open( unit = 2,  file = 'look')

c

c   ************************************************************
            sumerror = 0
c *************************************************************
c load in values from 'smith.all'.
c   give initial values of a and b

         alfa(1) = 130
         beta(1) = 60
         alfa(2) = 60
         beta(2) = 30
         alfa(3) = 45
         beta(3) = 20
         alfa(4) = 50
         beta(4) = 20
c
         do 200 m = 1,4



c reset all variables to zero

         olderror = 0
         newerror = 0
         deltaea1 = 0
         deltaea2 = 0
         deltaeb1 = 0
         deltaeb2 = 0
         incrementa = 0
         incrementb = 0
         suma = 0
```

```
        sumb = 0
        deda = 0
        dedb = 0
        count = 0

c load in the modes

        write(2,*)  'i the number of bins =' ,i,'*********'
        write(2,*)
        do 10 n = 1,22
            read(10 + m,*) d(m,n), cn(m,n)
c             write(*,*) n, cn(m,n)
10      continue
        write(2,*)



        a = alfa(m)
        b = beta(m)
c
c ******************************************************
c
c Calculate cr

        do 20 n = 1,i
              r = d(m,n)
            cr(m,n) = (1 - (r/a)**2)*exp(-(r/b)**2)

20      continue

c
c *********************************************************
c
c calculate value of error ( that is to be minimized)
c
         do 30 n = 1,i
            error(m,n) = (cr(m,n) - cn(m,n))**2
            sumerror = sumerror + error(m,n)
30       continue
         olderror = sumerror/n
c


c
c *****************************************************
c
c   calculate the delta error,and decide whether
c to add or subtract the increment

       deltaa = 0.001
       deltab = 0.001
c
           suma = 0
          sumb = 0
```

```
          do 50 n = 1,i
         deda = (4*(r**2)*exp(-(r/b)**4)/(a**3))*(1-(r/a)**2)
     $       - 4*(r**2)*(exp(-(r/b)**2))*cn(m,n)/(a**3)
             suma = suma + deda
50        continue

          deda = suma/n

          deltaea1 =  deda *  deltaa
          deltaea2 =  deda * (- deltaa)

            if (deltaea1.lt.deltaea2)  then
              incrementa = deltaa
              else
              incrementa = -deltaa
             end if

c         write(*,*)  deda,deltaea1,deltaea2,incrementa

         do 60 n =1,i
            dedb = (4*(r**4)*(exp(-(r/b)**4))/b**5)
     $           * (1-2*(r/a)**2 + (r/a)**4)
     $           + (4*(r**2)*cn(m,n)*(exp(-(r/b)**2))/b**3)
     $            * ((r/a)**2 -1)

           sumb = sumb + dedb
60       continue

          dedb = sumb/n

           deltaeb1 =    dedb * deltab
           deltaeb2 =    dedb *(- deltab)

             if (deltaeb1.lt.deltaeb2) then
                incrementb = deltab
                else
                incrementb = -deltab
            end if
c            write(*,*)  dedb,deltaeb1,deltaeb2,incrementb

c
c
c *************************************************************
c *************************************************************
c
c      incremant a and b.
c

100              count = count + 1
              a = a + incrementa      ! normally + -  - +
              b = b - incrementb      ! normally - -  + +
c
c ***********************************************************
c
```

```
c     calculate cr again
c
         do 70  n  = 1,i
             r = d(m,n)
             cr(m,n) = (1 - (r/a)**2)*exp(-(r/b)**2)
70       continue
c
c ****************************************************
c
c    calculate error again. with new a and b
c
            sumerror = 0
           do 80 n = 1,i
             error(m,n) = (cr(m,n) - cn(m,n))**2
             sumerror = sumerror + error(m,n)
80        continue
          newerror = sumerror/n

          write(*,*) m,count,newerror,a,b
c           stop
c ***********************************************************
c
        if (newerror.gt.olderror) then
       write(2,*)
       write(2,*)  'for minimized error, mode',m
       write(2,*)'     a             b        error       iterations'
       write(2,*)   a,b,newerror,count
       write(2,*)
            goto 195
            else
            olderror = newerror
            goto 100
         end if
195      alfa(m) = a
         beta(m) = b
200     continue
         call  corro(alfa,beta,i)
         close(11)
         close(12)
         close(13)
         close(14)
        end
```

```
      subroutine corro(alfa,beta,i)
       parameter(i = 8)
      integer m,n
      real alfa(4),beta(4),a,b,cr(4,i),bin,r


       open(unit = 11,file ='cor1')
       open(unit = 12,file ='cor2')
       open(unit = 13,file ='cor3')
       open(unit = 14,file ='cor4')

c      alfa(1) = 115.37
c      beta(1) =  90.37
c      alfa(2) = 84.18
c      beta(2) = 59.18
c      alfa(3) = 145.46
c      beta(3) = 45.42
c      alfa(4) = 51.86
c      beta(4) = 46.86

       bin = 25
c ***********************************************************
c
c     calculate cr
c
        do 60 m =1,4
           a = alfa(m)
           b = beta(m)
           write(*,*) alfa(m),beta(m)

        do 70  n  = 1,i
            r =  n * bin
            cr(m,n) = (1 - (r/a)**2)*exp(-(r/b)**2)
c  print out
         write(10 + m ,*) r,cr(m,n)
70      continue
60      continue


        end
```

```
c        APPENDIX 5B

C     SPACE-TIME OBJECTIVE ANALYSIS/STATISTICAL FORECAST PACKAGE
C     USING GENERALIZED OBJECTIVE ANALYSIS ROUTINE
C
C              (C) COPYRIGHT EVERETT CARTER 1981
C
C       uses NCAR double precision matrix inverter INVMTX
C
C     UPDATES:
C              21 Aug 1984 -- Modified package so that GETAVE is called
C                                within OBJAN, also added COMMON block CBLOCK
C                                in order to reduce correlation function calls
C               8 Aug 1984 -- added routine GETAVE, to remove weighted mean
C              27 DEC 1983 -- expanded IER flags
C               3 NOV 1983 -- added poor matrix inversion Warning
C
C     IER     is an error flag for OBJAN
C           =0      for no errors detected
C           >0      for matrix inversion errors (see matrix inversion routine)
C           =-1     for no input data (a Warning-- not fatal)
C           =-3     for poor matrix inversion, it did it but the inversion was
C                       nearly NUMERICALLY singular
C
C
C
C      THE MAIN PROGRAM MUST SET UP THE DIMENSIONS AS FOLLOWS
C          (FOR A 33X33 FIELD)
C     DIMENSION PSI(1089),XOBS(1089,2),TOBS(1089)
C     DIMENSION X(2,1089),THETA(1089),EPS(1089)
C     DIMENSION PARSI(20),T(20),PARX(2,20)
C     COMMON BLOCK ERR CONTAINS THE OBSERVATION ERROR PARAMETERS
C     E IS THE MEAN SQUARE NOISE LEVEL IN TERMS OF PERCENT OF VAR
C     COMMON/ERR/E
C     THE FUNCTION F IS THE CORRELATION FUNCTION
C     EXTERNAL F
C     M IS THE TOTAL NUMBER OF GRID POINTS
C     LIMIT IS THE MAXIMUM NUMBER OF INFLUENTIAL POINTS
C     DATA LIMIT/10/
C     DATA M/1089/
C     DATA DIST,TIM/100.,20./
C     PSI   THE OBSERVATION VALUES
C     XOBS  THE OBSERVATION POSITIONS
C     TOBS  THE OBSERVATION TIMES
C     TCEN  THE CENTRAL INTERPOLATION TIME (PREDICTION TIME)
C     X     THE INTERPOLATION POSITIONS
C     THETA THE INTERPOLATION VALUE OF THE COMPLETED FIELD
C     EPS   THE INTERPOLATION ERROR LIMIT OF THE COMPLETED FIELD
C     E=0.05
C     EXAMPLE MAIN LOOP
C     DO 150 KX=1,M
C        CALL SELECT(LIMIT,X(KX,1),X(KX,2),TCEN,XOBS,TOBS,PSI,
C    1                PARSI,PARX,T,N,NOBS,DIST,TIM)
```

```
C         CALL OBJAN(PARSI,PARX,T,NOBS,X(KX,1),X(KX,2),
C    1                TCEN,B,W,IER)
C         THETA(KX)=B+AVER
C         EPS(KX)=W/VAR
C 150 CONTINUE
C
C
C
      SUBROUTINE REMAV(PSI,M,AVER,SDV)
C     ROUTINE TO CALCULATE THE MEAN AND VARIANCE OF AN ARRAY
C     IT ALSO REMOVES THE MEAN FROM THE ARRAY
      DIMENSION PSI(1)
      AVER=0.
      SDV=0.
      DO 10 I=1,M
         AVER=AVER+PSI(I)
         SDV=SDV+PSI(I)**2
   10 CONTINUE
      AVER=AVER/FLOAT(M)
      SDV=SDV/FLOAT(M)-AVER**2
      IF (M .NE. 1) SDV=(FLOAT(M)/FLOAT(M-1))*SDV
      DO 20 I=1,M
         PSI(I)=PSI(I)-AVER
   20 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SELECT(LIMIT,X,Y,TCEN,XOBS,T,PSI,
     1                   PARSI,PARX,TOBS,N,NOBS,DIST,TIM,alfa,beta)
C     ROUTINE TO SELECT THE AT MOST "LIMIT" NEARBY POINTS
C     TO AN INTERPOLATION POINT X,Y,TCEN
C     LIMIT IS THE MAXIMUM NUMBER OF POINTS TO USE
C     DIST IS THE SPATIAL RADIUS OF INFLUENTIAL POINTS IN KM
C     TIM IS THE TEMPORAL RADIUS OF INFLUENTIAL POINTS IN DAYS
      DIMENSION XOBS(2,1),T(1),PSI(1),PARSI(1),TOBS(1)
      DIMENSION PARX(2,1)
      DIMENSION INDEX(2000),COR(2000)
      real a,b
      COMMON/CBLOCK/C(20)
      EXTERNAL F
      DATA CPHSE/0.0/
      NOBS=0
      DO 50 I=1,N
         DELX=X-XOBS(1,I)
         DELY=Y-XOBS(2,I)
         DELT=TCEN-T(I)
         R=SQRT((DELX-CPHSE*DELT)**2+DELY**2)
         IF (ABS(DELT) .GT. TIM) GOTO 50
         IF (R .GT. DIST) GOTO 50
         NOBS=NOBS+1
         INDEX(NOBS)=I
         COR(NOBS)=F(DELX,DELY,DELT,alfa,beta)
```

```
   50 CONTINUE
      IF (NOBS .EQ. 0) GOTO 75
      IF (NOBS .GT. LIMIT) CALL SORT(COR,INDEX,NOBS)
      IF (NOBS .GT. LIMIT) NOBS=LIMIT
      DO 70 I=1,NOBS
         J=INDEX(I)
         PARX(1,I)=XOBS(1,J)
         PARX(2,I)=XOBS(2,J)
         TOBS(I)=T(J)
         PARSI(I)=PSI(J)
         C(I)=COR(I)
   70 CONTINUE
   75 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SORT(COR,INDEX,N)
C     A SHELL SORT ROUTINE TO SORT INDEX AND COR DOWN
C     ACCORDING TO THE VALUES OF COR
      DIMENSION COR(1),INDEX(1)
      IGAP=N
    5 IF (IGAP .LE. 1) RETURN
      IGAP=IGAP/2
      IMAX=N-IGAP
   10 IEX=0
      DO 20 I=1,IMAX
         IPLUSG=I+IGAP
         IF (COR(I) .GE. COR(IPLUSG)) GOTO 20
         SAVE=COR(I)
         COR(I)=COR(IPLUSG)
         COR(IPLUSG)=SAVE
         ISAVE=INDEX(I)
         INDEX(I)=INDEX(IPLUSG)
         INDEX(IPLUSG)=ISAVE
         IEX=1
   20 CONTINUE
      IF (IEX .NE. 0) GOTO 10
      GOTO 5
      END
C
C
C
      SUBROUTINE OBJAN(PSI,L,T,N,X,Y,TCEN,B,W,IER,alfa,beta)
C     THE SPACE-TIME OBJECTIVE ANALYSIS ROUTINE
C     VERSION FOR 1 INTERPOLATION POINT
C     USES 2 SPACE AND 1 TIME DIMENSION
C     NOTE      DELTA T = TCEN - T(J)
C     L IS THE ARRAY OF OBSERVATION POSITIONS, IN KM
C     T IS THE TIME OF OBSERVATION, IN DAYS
C     X IS THE ARRAY OF INTERPOLATION POSITIONS, IN KM
C     TCEN IS THE CENTRAL INTERPOLATION TIME
C     PSI IS THE ARRAY OF OBSERVATION VALUES
```

```
C     B IS THE INTERPOLATED VALUE
C     W IS THE INTERPOLATION ERROR LIMIT
C     N IS THE NUMBER OF OBSERVATION POINTS
C     IER IS AN ERROR FLAG, ZERO FOR NO ERROR
C                               -1     No data (WARNING)
C                               -3     Poor matrix inversion (WARNING)
      DIMENSION PSI(1),T(1),L(2,1)
      COMMON/CBLOCK/C(20)
      REAL*8 A(20,20)
      REAL L
      COMMON/ERR/E
      EXTERNAL F
      B=0.
      W=1.0
      IER=-1
      IF (N .LE. 0) GOTO 500
C      CALCULATE THE INVERTED AUTOCORRELATION MATRIX FOR THE OBSERVATIONS
      CALL SETA(A,L,T,N,IER,alfa,beta)
      IF (IER .GT. 0) GOTO 500
      CALL GETAVE(A,PSI,N,AVE)
C     CALCULATE THE MATRIX C
C     -- already calculated in this version, common block CBLOCK
C
C     CALCULATE THE RMS INTERPOLATION ERROR,W
C     AND CALCULATE THE INTERPOLATED VALUE B
      W=0.
      B=0.0
      DO 150 I=1,N
         H=0.0
         DUMC=C(I)
         DO 140 J=1,N
            P=DUMC*C(J)*SNGL(A(I,J))
            W=W+P
            P2=SNGL(A(I,J))*PSI(J)
            H=H+P2
  140    CONTINUE
         DUMY=DUMC*H
         B=B+DUMY
  150 CONTINUE
      B=B+AVE
      W=ABS(1.-W)
  500 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE SETA(A,PARX,T,NOBS,IER,alfa,beta)
C     THIS ROUTINE CALCULATES THE AUTOCORRELATION MATRIX FOR THE
C     OBSERVATIONS GIVEN THE POSITIONS, PARX AND TIMES, T
C     IT RETURNS THE INVERTED MATRIX
      DIMENSION PARX(2,1),T(1)
      REAL*8 A(20,20),Det
      Integer IP(40)
```

```
      COMMON/ERR/E
      EXTERNAL F
      DATA NA/20/
C     The Guard value for DETERMINANT WARNINGS
      DATA GUARD/1.0E-4/
    1 FORMAT (5X,'MATRIX A IS SINGULAR')
    2 FORMAT (5X,'ERROR,MATRIX A IS TOO SMALL',/,
     X             ' NA MUST BE .GE. NOBS',/,' NA=',I3,5X,'NOBS=',I3,//)
    3 FORMAT (5X,'WARNING, DETERMINANT IS VERY SMALL (',1PE11.4,')',
     X          ' -- TRY SMALLER NUMBER OF INFLUENTIAL POINTS')
C     TEST THE SIZE OF THE OBSERVATION ARRAY
      IER=1
      IF (NA .LT. NOBS) PRINT 2,NA,NOBS
      IF (NA .LT. NOBS) RETURN
      IER=0
      DO 20 I=1,NOBS
         DO 10 J=I,NOBS
            DELT=T(I)-T(J)
            DELX=PARX(1,I)-PARX(1,J)
            DELY=PARX(2,I)-PARX(2,J)
            A(I,J)=DBLE(F(DELX,DELY,DELT,alfa,beta))
            A(J,I)=A(I,J)
   10    CONTINUE
       A(I,I)=A(I,I)+DBLE(E)
   20 CONTINUE
C     INVERT THE NOBS*NOBS MATRIX A
C     THE INVERTED MATRIX IS NAMED A
      Call InvMtx(A,NA,A,NA,NOBS,Det,IP,Ier,alfa,beta)
      IF (IER .NE. 0) PRINT 1
      IF (IER .NE. 0) GOTO 40
C     CHECK THE DETERMINANT
      IF (DET .LT. GUARD) PRINT 3,DET
      IF (DET .LT. GUARD) IER=-3
   40 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE GETAVE(A,PSI,N,AVE)
C     Calculate and remove the weighted mean
      DIMENSION PSI(1)
      DIMENSION C(20),D(20)
      REAL*8 A(20,20)
      DO 10 I=1,N
         C(I)=0
         D(I)=0
         DO 10 K=1,N
            C(I)=C(I)+A(I,K)*PSI(K)
            D(I)=D(I)+A(I,K)
   10 ENDDO
      SUM1=0
      SUM2=0
      DO 20 I=1,N
```

```
      SUM1=SUM1+C(I)
      SUM2=SUM2+D(I)
20 ENDDO
   AVE=SUM1/SUM2
   DO 30 I=1,N
         PSI(I)=PSI(I)-AVE
30 ENDDO
   RETURN
   END
```

```
c     APPENDIX 6B

c this is the program that links all the elements together.
c and also controls the reduction or removal of successive bathys.
c
        parameter(most = 56)
        real alfa(4),beta(4),amode(most),xpos(most),ypos(most)
        real a,b
        integer n,printer,loop,ran(most),i,j,left,ans,num ,reply


c read in the data for each time data set is reduced by one
c
         do 100 z = 1,most
          write(*,*) ' process 1 = y 2 = end'
         read(*,*)   reply
        if(reply.eq.2) goto 200

         write( *,*) 'how many XBTs do you want to use ( max 133)'
         read(*,*) ans

           j = most - ans

        call reduce(j)
        left = most - j
c
c read in modes
       do 10 n = 1,4
       open ( unit =  1, file = 'rdecpos.mat')
       open ( unit = 11, file = 'rmod1')
       open ( unit = 12, file = 'rmod2')
       open ( unit = 13, file = 'rmod3')
       open ( unit = 14, file = 'rmod4')
c
c
c set parameters for input
c

c chose values for a and b
c
         alfa(1) = 155.5
         beta(1) =  85.3
         alfa(2) =  92.15
         beta(2) =  62.14
         alfa(3) =  36.65
         beta(3) =  28.35
         alfa(4) =  39.4
         beta(4) =  30.55
c

c
c ************************************************
c
```

```
c now loop through oa four times.once for each mode
c


c read in latitude and longitude of obs
c
      do 26 i = 1,left
      read(1,*) num,xpos(i),ypos(i)
26    continue
      rewind(1)
c
c read in modes
           do 27 i = 1,left
              read(10 + n,*) amode(i)
27         continue

         close(1)
         close(11)
         close(12)
         close(13)
         close(14)
c
c give  values of a and b
c
             a = alfa(n)
             b = beta(n)

c set output diagnostics.
c
             printer = 30 + n
             loop = n
c

c ***********************************************************
c
             call modeoa(amode,xpos,ypos,a,b,printer,left,loop)


10       continue

c print out modes and errors to combined files


          call allmods

100        continue

200        write(*,*) 'program finished'
            end
```

```
c       Program by E.F. CARTER

      subroutine modeoa(amode,x,y,alfa,beta,printer,most,loop)
c       PROGRAM modeoa
C
C
C
C     UNIT 1  IS THE XBT (INPUT) DATA
C     UNIT 2  IS THE PRINTABLE OUTPUT DATA (DIAGNOSTICS)
C     UNIT 4  IS THE UNFORMATTED OUTPUT DATA
C
c       INTEGER INFILE,DISK,PRINTER
c     PARAMETER (INFILE=1,  DISK=4)
C
      Parameter (MXOBS = 56)
      INTEGER DAY(MXOBS),Gmt(MXOBS),printer,most
      DIMENSION X(MXOBS),Y(MXOBS),Tin(MXOBS),amode(MXOBS)
      DIMENSION XOBS(2,MXOBS),TOBS(MXOBS),UOBS(MXOBS)
      DIMENSION UOPT(20),VOPT(20),TOPT(20),XOPT(2,20)
      DIMENSION XI(119),YI(119),UI(119),ERRU(119)
      INTEGER START,EMPTY,loop
      Real MnDay,MxDay,alfa,beta
      COMMON/ERR/E
      EXTERNAL F
      DATA EMPTY/156/
c      DATA MOST/57/
      DATA LIMIT/5/
C     SPATIAL AND TEMPORAL LIMITS
      DATA DIST,TIM/150.,0./
      DATA CFX,CFY/15.0,15.0/
      DATA START/1/
      DATA TINC/1/
C
      DATA NOBJ/1/
c
c *************************************************************

c       write(*,*) 'modeoa now being called'

c open files for output
       open (unit = 31, file = 'output1')
       open (unit = 41, file = 'gridmod.1')
       open (unit = 51, file = 'errormod.1')
       open (unit = 32, file = 'output2')
       open (unit = 42, file = 'gridmod.2')
       open (unit = 52, file = 'errormod.2')
       open (unit = 33, file = 'output3')
       open (unit = 43, file = 'gridmod.3')
       open (unit = 53, file = 'errormod.3')
       open (unit = 34, file = 'output4')
       open (unit = 44, file = 'gridmod.4')
       open (unit = 54, file = 'errormod.4')
c *************************************************************
```

```
c
    1 FORMAT (5X,'DIAGNOSTICS OF ALL modal OBSERVATIONS :')
    2 Format (5x,'X Position Diagnostics (Km): ')
    3 Format (5x,'Y Position Diagnostics (Km): ')
   22 Format (5x,'X grid Diagnostics (Km): ')
   23 Format (5x,'Y grid Diagnostics (Km): ')
    4 FORMAT (5X,'INTERPOLATED mode FIELD DIAGNOSTICS:')
    5 FORMAT (5X,'INTERPOLATED mode Error FIELD DIAGNOSTICS:')
    6 FORMAT (5X,'JULIAN DATE:',F8.3,
     &             ' NUMBER OF Observations used :',I4)
    7 FORMAT (5X,'MINIMUM DATE:'F8.3,' MAXIMUM DATE:',F8.3)
    8 FORMAT (7X,'ERROR, XOBS TOO SMALL.  MXOBS=',I3)
    9 FORMAT (5X,'Assumed NOISE LEVEL:',F8.5)
   10 Format (5X,'Date (Time) Diagnostics (Julian Date): ')
   18 FORMAT (5X,'Number of influential points is : ',I4,//)
c
c ***********************************************************
c
      E=0.1
      WRITE (PRINTER,9) E
c
c ***********************************************************

C     READ IN THE OBSERVATION DATA
c       I=1
c   25 Continue
c       OPEN (UNIT=INFILE,File='xbt.dat')
c       READ (INFILE,*,End=30) Day(I),Gmt(I),Y(I),X(I),SST(I)
c       I=I+1
c       Goto 25
c   30 Continue
c     Most=I-1
c       CLOSE (UNIT=INFILE)
c ************************************************************
c
c Read in the observed data.

c for day and time
       do 25 i = 1,most
              Day(i) = 1
              Gmt(i) = 1
              tin(i) = 1
25     continue

c for latitude and longitude of observations
c
c        open (unit = 1, file = 'dec.pos')
       do 26 i = 1,most
           write(*,*) x(i),y(i)
26     continue
c
c  for modal amplitudes
c
c      open ( unit = 2, file = 'xaa')
```

```
c       do 27 i = 1,most
c           read(2,*) amode(i)
c27     continue
c
c    for latitude and longitude of grid points
c
      open(unit = 3, file = 'grid.pos')
       do 28 i = 1,119
           read(3,*) n,xi(i),yi(i)
28       continue
         rewind(3)
         close(3)
c *************************************************************


      CALL SCALE(X,Y,Most)


      Write (Printer,2)
      Call Diag(X,Most,Printer)
      Write (Printer,3)
      Call Diag(Y,Most,Printer)
c
      CALL SCALE(Xi,Yi,119)
c
      Write (Printer,22)
      Call Diag(Xi,Most,Printer)
      Write (Printer,23)
      Call Diag(Yi,Most,Printer)
c
c     CALL JULIAN(DAY,Gmt,Tin,Most)
c
      Write (Printer,10)
      Call Diag(Tin,Most,Printer)
c
      WRITE (PRINTER,1)
      CALL DIAG(amode,MOST,PRINTER)
c
      Call Remav(amode,Most,Ave,V)
c
c *******************************************************
C     SET UP THE INTERPOLATION POSITIONS
c      M=0
c      DO 40 J=1,21
c         DO 40 I=1,33
c            M=M+1
c            XI(M)=CFX*(I-17)
c            YI(M)=CFY*(J-11)
c   40 ENDDO
c
c ***********************************************************

C     DO SEVERAL ANALYSES
C
```

```
      WRITE (PRINTER,18) LIMIT
      TCEN=START-TINC
      DO 500 IOBJ=1,NOBJ
         TCEN=TCEN+TINC
C     GET THE USABLE OBSERVATIONS FOR THIS DATE
         CALL GETOBS(Tin,X,Y,amode,Most,TCen,Tim,
     X               Tobs,Xobs,UOBS,N)
        MNDAY=TCen-TIM
        MXDAY=TCen+TIM
        WRITE (PRINTER,6) Tcen,N
        WRITE (PRINTER,7) MNDAY,MXDAY
        WRITE (PRINTER,18) LIMIT
        IF (N .EQ. 0) GOTO 500
        IF (N .GT. MXOBS) THEN
               WRITE (PRINTER,8) MXOBS
               GOTO 500
        ENDIF
C
C   m is number of grid positions,and will not change.

            m = 119

           DO 50 k =1, m

           CALL SELECT(LIMIT,XI(K),YI(K),TCEN,XOBS,TOBS,UOBS,
     X                 UOPT,XOPT,TOPT,N,NOBS,DIST,TIM,alfa,beta)

C           Print *,'N, Nobs, ',N,Nobs
C           Do 49 IPXJ=1,NOBS
C              Print *,Xopt(1,IPXJ),XOpt(2,IPXJ),Uopt(IPXJ)
C    49     EndDo
           CALL OBJAN(UOPT,XOPT,TOPT,NOBS,XI(K),YI(K),
     X          TCEN,UI(K),ERRU(K),IER,alfa,beta)
           UI(K)=UI(K)+Ave
  50     ENDDO

c
         WRITE (PRINTER,4)
         CALL DIAG(UI,M,PRINTER)
         WRITE (PRINTER,5)
         CALL DIAG(ERRU,M,PRINTER)


         WRITE (30 + loop,*)  'centre time',TCen
         WRITE (30 + loop,*)  ' number of obs points',N

         WRITE (50 + loop,*)  -72, -64, 37 ,40
         write (50 + loop,*)  17, 7
         WRITE (40 + loop,*)  -72, -64, 37 ,40
         write (40 + loop ,*)  17, 7
         do 51 i = 1,m
         WRITE (40 +loop,*) UI(i)
         WRITE (50 +loop,*) ERRU(i)
51       continue
```

```
 500 CONTINUE
        close(31)
        close(32)
        close(33)
        close(34)
        close(41)
        close(42)
        close(43)
        close(44)
        close(51)
        close(52)
        close(53)
        close(54)
      END

c ****************************************************
c ****************************************************
c ****************************************************


      SUBROUTINE GETOBS(Day,Posx,Posy,UOBS,Ninp,CDay,Tim,T,X,U,N)
C
C     Input data:
C               Day,Posx,Posy   space-time location of data
C               UOBS            observed data
C               Ninp            number of input points
C               CDay            Central day of estimate
C               TIM             width of time window
C
C     Output data:
C               T,X             location of data
C               U               chosen observation data
C               N               number of points used
C
C     ROUTINE TO GET THE OBSERVATION DATA
C     N IS THE NUMBER OF OBSERVATIONS
C
      DIMENSION T(1),X(2,1),U(1)
      DIMENSION Day(1),UOBS(1),POSX(1),POSY(1)
      Real MXDAY,MNDAY
      MXDAY=CDAY+TIM
      MNDAY=CDAY-TIM
      KOUNT=0
      Do 10 I=1,Ninp
         IF (DAY(I) .GT. MXDAY) GOTO 10
         IF (DAY(I) .LT. MNDAY) GOTO 10
         KOUNT=KOUNT+1
         T(KOUNT)=DAY(I)
         X(1,KOUNT)=POSX(I)
         X(2,KOUNT)=POSY(I)
         U(KOUNT)=UOBS(I)
  10  Continue
      N=KOUNT
      RETURN
```

```
      END
C
C
C
      SUBROUTINE SCALE(X,Y,N)
C     Scale Lat and Long to Km
      DIMENSION X(1),Y(1)
      Parameter (XCen = -70.5, YCen = 38.25)
      Parameter (CFX = 110.99, CFY = 87.84)
      DO 10 I=1,N
         Y(I)=CFY*( Y(I)-YCen)
         X(I)=CFX*( X(I)-XCEN)
   10 ENDDO
      RETURN
      END
C
C
C
      SUBROUTINE JULIAN(DAY,Gmt,Time,N)
      INTEGER DAY(1),Gmt(1)
      Dimension Time(1)
      Integer Offset
      Parameter (Offset = 86000, Julian86 = 6431)
      Real MnDay,MxDay,Minutes
    1 FORMAT (5X,'MIN DATE :',F8.3,' MAX DATE :',F8.3)
      MNDAY=9999.9
      MXDAY=0.0
      DO 10 I=1,N
         Time(I)=Float(Gmt(I))/100.
         Minutes=100.*(Time(I) - IFix(Time(I)))
         Time(I)=Time(I) + Minutes/60.0
         Time(I) = Float(Julian86 + Day(I) - Offset) + Time(I)/24.0
         IF (Time(I) .GT. MXDAY) MXDAY=Time(I)
         IF (Time(I) .LT. MNDAY) MNDAY=Time(I)
   10 ENDDO
      WRITE (2,1) MNDAY,MXDAY
      RETURN
      END
C
C
C
      FUNCTION F(X,Y,T,alfa,beta)
C     THE CORRELATION FUNCTION
C     THE SCALE FACTORS
c      Parameter (a=111.6, b=86.6)
      a = alfa
      b = beta
      r2=x**2 + y**2
      F=(1.0 - r2/a**2)*exp(-r2/b**2)
      RETURN
      END
```

```
C this program generates first 158  numbers ranomly
c
       parameter(max = 58)
      integer i,m,x(max),n

       open( unit =1, file = 'randnumdeep')
       call srand(3)
       do 10 i = 1 , max
40           n = irand()

             x(i) = n

       if(i.gt.1) then
           do 20 m = 1,i-1
               if ((x(i).eq.x(m)).or.(x(i).gt.max-2)) goto 40
20         continue
       endif


           write(*,*) i,x(i)
           write(1,*) x(i)
10    continue
      end
```

```
close(1)
close(2)
close(11)
close(12)
close(13)
close(14)
close(20)
close(21)
close(22)
close(23)
close(24)
end
```

```
C      APPENDIX 7B

C     Program by E.F. Carter

      SUBROUTINE INVMTX (A,NA,V,NV,N,D,IP,IER,alfa,beta)
C     Double Precision version
C      MATRIX INVERSION     V=INV(A)
C     THE ARRAY A MAY BE ENTERED AS V TO SAVE MEMORY
C     IP MUST BE DIMENSIONED TO AT LEAST 2*N
      INTEGER NA,NV,N,IP(1),IER
      REAL*8             A(NA,N)     ,V(NV,N)     ,D
      Real*8     VMax, VH, PVT, PVTMX, HOLD
C     IEXMAX IS SET TO THE LARGEST BASE TEN EXPONENT THAT CAN BE
C     REPRESENTED ON THE MACHINE,  I.E. LARGEST=10**IEXMAX
      DATA IEXMAX/38/
  115 FORMAT(28H0*MATRIX SINGULAR IN INVMTX*)
  116 FORMAT(34H0*DETERMINANT TOO LARGE IN INVMTX*)
      IER = IERINV(N,NA,NV)
      IF (IER .NE. 0) RETURN
      DO 102 J=1,N
         IP(J) = 0
         DO 101 I=1,N
            V(I,J) = A(I,J)
  101    CONTINUE
  102 CONTINUE
      D = 1.
      IEX = 0
      DO 110 M=1,N
         VMAX = 0.
         DO 104 J=1,N
            IF (IP(J) .NE. 0) GO TO 104
            DO 103 I=1,N
               IF (IP(I) .NE. 0) GO TO 103
               VH = ABS(V(I,J))
               IF (VMAX .GE. VH) GO TO 103
               VMAX = VH
               K = I
               L = J
  103       CONTINUE
  104    CONTINUE
         IP(L) = K
         NPM = N+M
         IP(NPM) = L
         D = D*V(K,L)
  105    IF (ABS(D) .LE. 1.0) GO TO 106
         D = D*0.1
         IEX = IEX+1
         GO TO 105
  106    CONTINUE
         PVT = V(K,L)
         IF (M .EQ. 1) PVTMX = ABS(PVT)
         IF (ABS(PVT/FLOAT(M))+PVTMX .EQ. PVTMX) GO TO 113
         V(K,L) = 1.
         DO 107 J=1,N
```

```
        subroutine reduce(j)
        parameter(most = 56)
        integer i,j,ran(most),val,p
        real amod1(most),amod2(most),amod3(most)
        real amod4(most),xpos(most),ypos(most)

c read in  original modes

       open (unit = 2, file = 'randnumdeep')

       open ( unit =  1, file = 'decpos.mat')
       open ( unit = 11, file = 'mod1.mat')
       open ( unit = 12, file = 'mod2.mat')
       open ( unit = 13, file = 'mod3.mat')
       open ( unit = 14, file = 'mod4.mat')
c
c write out reduced set.
       open ( unit = 20, file = 'rdecpos.mat')
       open ( unit = 21, file = 'rmod1')
       open ( unit = 22, file = 'rmod2')
       open ( unit = 23, file = 'rmod3')
       open ( unit = 24, file = 'rmod4')
c
c *********************************************************
c
c   this section will reduce data set by j,the parameter fed from
c  program driver.


           p =0
        do 100 i = 1,most
              read(2,*)  ran(i)
              write(*,*) i, ran(i)
              read(1,*)  xpos(i),ypos(i)
              read(11,*) amod1(i)
              read(12,*) amod2(i)
              read(13,*) amod3(i)
              read(14,*) amod4(i)
100      continue
c

           do 110 i = 1,most
              do 120 n = 1,j
                if(i.ne.ran(n))  goto 120
                if(i.eq.ran(n)) goto 110
120           continue

                 p = p + 1
                write(20,*) p,xpos(i),ypos(i)
                write(21,*) amod1(i)
                write(22,*) amod2(i)
                write(23,*) amod3(i)
                write(24,*) amod4(i)
110          continue
```

```
            HOLD = V(K,J)
            V(K,J) = V(L,J)
            V(L,J) = HOLD/PVT
107      CONTINUE
         DO 109 I=1,N
            IF (I .EQ. L) GO TO 109
            HOLD = V(I,L)
            V(I,L) = 0.
            DO 108 J=1,N
               V(I,J) = V(I,J)-V(L,J)*HOLD
108         CONTINUE
109      CONTINUE
110   CONTINUE
      M = N+N+1
      DO 112 J=1,N
         M = M-1
         L = IP(M)
         K = IP(L)
         IF (K .EQ. L) GO TO 112
         D = -D
         DO 111 I=1,N
            HOLD = V(I,L)
            V(I,L) = V(I,K)
            V(I,K) = HOLD
111      CONTINUE
112   CONTINUE
      IF (IEX .GT. IEXMAX) GO TO 114
      D = D*10.**IEX
      RETURN
113   IER = 33
      PRINT 115
      RETURN
114   IER = 1
      D = FLOAT(IEX)
      PRINT 116
      RETURN
      END
      FUNCTION IERINV (N,NA,NV)
103   FORMAT(23H0* N .LT. 1 IN INVMTX *)
104   FORMAT(24H0* NA .LT. N IN INVMTX *)
105   FORMAT(24H0* NV .LT. N IN INVMTX *)
      IERINV = 0
      IF (N .GE. 1) GO TO 101
      IERINV = 34
      PRINT 103
      RETURN
101   IF (NA .GE. N) GO TO 102
      IERINV = 35
      PRINT 104
      RETURN
102   IF (NV .GE. N) RETURN
      IERINV = 36
      PRINT 105
      RETURN
```

```
%       APPENDIX  8B

% this matlab file reconstructs the bathys at the grid points
% using 4 modes only, after the OA with reducing number of
% initial XBT's.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% this part reconstructs all synthetic, grid, XBTs using 4 modes.
%
clear
clg
hold off


load vec;
load gridmod;
load errmod;
grid = gridmod';
%
for j = 1:4;
for i = 1:80;
eigvec(i,j) = vec(i,(81-j));
end
end

    recbath = eigvec*grid;
    save  recbath.mat recbath /ascii;

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOT GIVEN XBT
% this part plots a given bathy, SELECT XBT NUMBER
% 1,  being lower left corner.


% maxis = [0 30 -80 0];
% axis(maxis);
%plot(recbath(56),'.');
% grid;
%title('  XBT 56 ');
%ylabel ('depth metres');
% xlabel('temp Celcius');
% hold on
% pause
%print
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% now to calculate the error in each reconstructed bathy
 eigvecsrd = eigvec' * eigvec;
  xbtvar = eigvecsrd*errmod';
  xbtvar = sum(xbtvar)
  view = reshape(xbtvar,17,7)

view = view';
```

```
view = flipud (view);
%view = nerr;
save batherr.mat view /ascii;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold off

% plot routine for MATLAB, of the reconstructed oa error.

for i = 1:17;
x(i) = -72.5 +0.5*i;
end
for i = 1:7;
y(i) = 36.5 + 0.5*i;
end

c = contour(view,x,y);
v = [0.2,0.4,0.6,0.8];
clabel(c,v);
grid;
% title('reconstruction error, using  1 out of 156  XBTs');    % the title must cha
xlabel('Longitude west from Greenwich');                   % each time
ylabel('Latitude North');
title(' reconsruction error variance using all 156 XBTs')

hold on


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
% loading in data positions.

load rdecpos

long = rdecpos(:,2);
lat  = rdecpos(:,3);
%plot(long,lat,'*');
%xlabel('Longitude west from Greenwich')
%ylabel('Latitude North')
%title ('position of deep XBTs')
%text(-70.5,38.5,'X')
%print
%pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% mark on box
box =[ -70.5   38.1
```

```
        -70.5   38.75
        -69.25   38.75
        -69.25   38.5
        -70.5  38.1]
blong = box(:,1)
blat  = box(:,2)
%plot(blong,blat)
pause
print
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% load in each modal error.

hold off
clg

moderr1 = errmod(:,1)';
moderr1 = reshape(moderr1,17,7);
moderr1 = moderr1';
moderr1 = flipud(moderr1);
k1 = contour(moderr1,x,y);
clabel(k1,v);
title ('mode 1 error')
%print
%pause

moderr2 = errmod(:,2)';
moderr2 = reshape(moderr2,17,7);
moderr2 = moderr2';
moderr2 = flipud(moderr2);
k2 = contour(moderr2,x,y);
clabel(k2,v);
title ('mode 2 error')
%print
%pause

moderr3 = errmod(:,3)'.
moderr3 = reshape(moderr3,17,7);
moderr3 = moderr3';
moderr3 = flipud(moderr3);
k3 = contour(moderr3,x,y);
clabel(k3,v);
title ('mode 3 error')
%print
%pause

moderr4 = errmod(:,4)';
moderr4 = reshape(moderr4,17,7);
moderr4 = moderr4';
moderr4 = flipud(moderr4);
k4 = contour(moderr4,x,y);
clabel(k4,v);
title ('mode 4 error')
%print
%pause
```

```
clg

subplot(221),contour(moderr1,x,y);
subplot(222),contour(moderr2,x,y);
subplot(223),contour(moderr3,x,y);
subplot(224),contour(moderr4,x,y);
%print
%clg


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% plot routine for CONTOUR
%nerr= nerr';
%save err.con nerr /ascii;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%
%
%pluserr = recbath(:,21) + totalerr(:,21);
%minuserr = recbath(:,21) - totalerr(:,21);
%plot(pluserr)
%plot(minuserr)
%glunk = totalerr(:,21);
%clong = recbath(:,21);
%save err7105.mat  glunk /ascii
%save bath7105.mat  clong /ascii
%for j = 1 : 1
%for i = 1:80
%deptherr(i,j) = glunk(i,j)*100/clong(i,j);
%end
%end
%save percenerr.mat deptherr /ascii
```

DISTRIBUTION LIST

Address	No of copies required
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5000	2
3. Chairman (Code OC/Co) Department of Oceanography Naval Postgraduate School Monterey, CA 93943-5000	1
4. Chairman (Code MR/Hy) Department of Oceanography Naval Postgraduate School Monterey, CA 93943-5000	1
5. Prof E.F. Carter (Code OC/Cr) Department of Oceanography Naval Postgraduate School Monterey, CA 93943-5000	1
6. Lt Cdr M.J. Sauze Royal Navy 1342 Castro Ct Monterey, CA 93940	1
7. Commanding Officer Fleet Numerical Oceanographic Center Monterey, CA 93943-5005	1
8. Hydrographer of the Navy Ministry of Defense Hydrographic Department Taunton, Somerset TA1 29N England	1

10. The Director of Naval Oceanography and Meteorology 1
8 th floor
Lacon House
Theobalds Road
London
WC1X 8RY England

11. Prof T. Rossby 1
Graduate School of Oceanography
222 Watkins Blg
University of Rhode Island
Narragansett, RI 02882

12. Officer in Charge 1
Fleet Oceanography Centre
Commander in Chief Fleet
HMS WARRIOR
Eastbury Park
Northwood
Middlesex
HA6 3HP
England

13. Officer in charge 1
Royal Naval School of
Meteorology and Oceanography
RNAS CULDROSE
Helston
Cornwall
England

14. Commanding Officer 1
Naval Oceanographic Office
Stennis Space Center
MS 39529-5001